

Um evento com propósito

# Embarcadero Conference 2020 Online

## TODOS CONECTADOS



Um evento com propósito  
**Embarcadero Conference**  
**2020 Online**

**EXPANDINDO O ALCANCE DO MULTIMÍDIA NO DELPHI:  
UMA APLICAÇÃO SMART TV  
EM RASPBERRY PI**

Jorge Manuel Lage Fernandes



# Sobre

- Jorge Manuel Lage Fernandes
- Bacharel em Ciências da Computação
- Mestre em Tecnologias de Gestão
- Empreendedor
- Membro da Universidade Delphi
- Membro da Comunidade da Sociedade de Engenharia de Televisão
- Apaixonado por Delphi / Object Pascal
  
- [jorge@youngarts.com.br](mailto:jorge@youngarts.com.br)
- [github.com/jmfjorge](https://github.com/jmfjorge)





AVALIE A PALESTRA

# Contextualizando

# Motivação

Curiosidade

Pesquisa & Desenvolvimento

Delphi + Áudio + Vídeo + Multiplataforma

Não usar componentes de terceiros

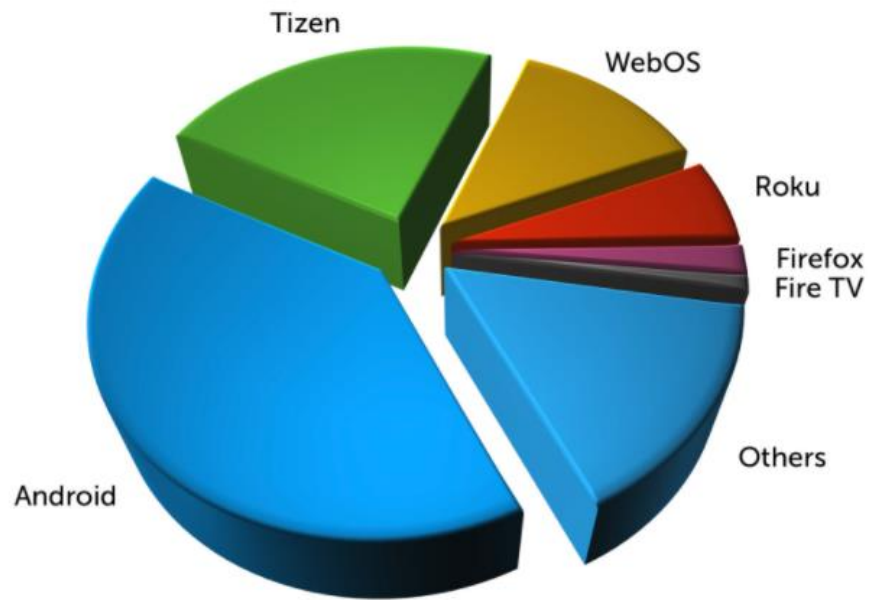
- Ex: FFmpeg / VLC (excelentes!)

Sair da caixa!

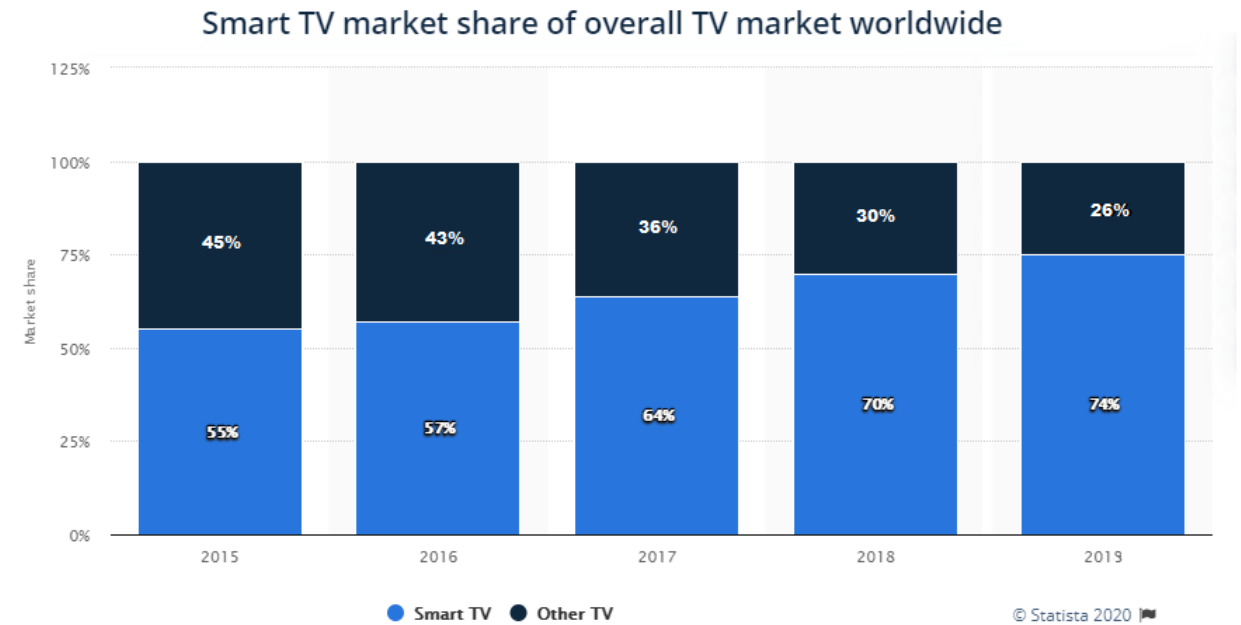
“É Delphi ?!”



# Motivação



Global Smart TV OS Market Share per Wakefield Research. Image: Digitized House



© Statista 2020





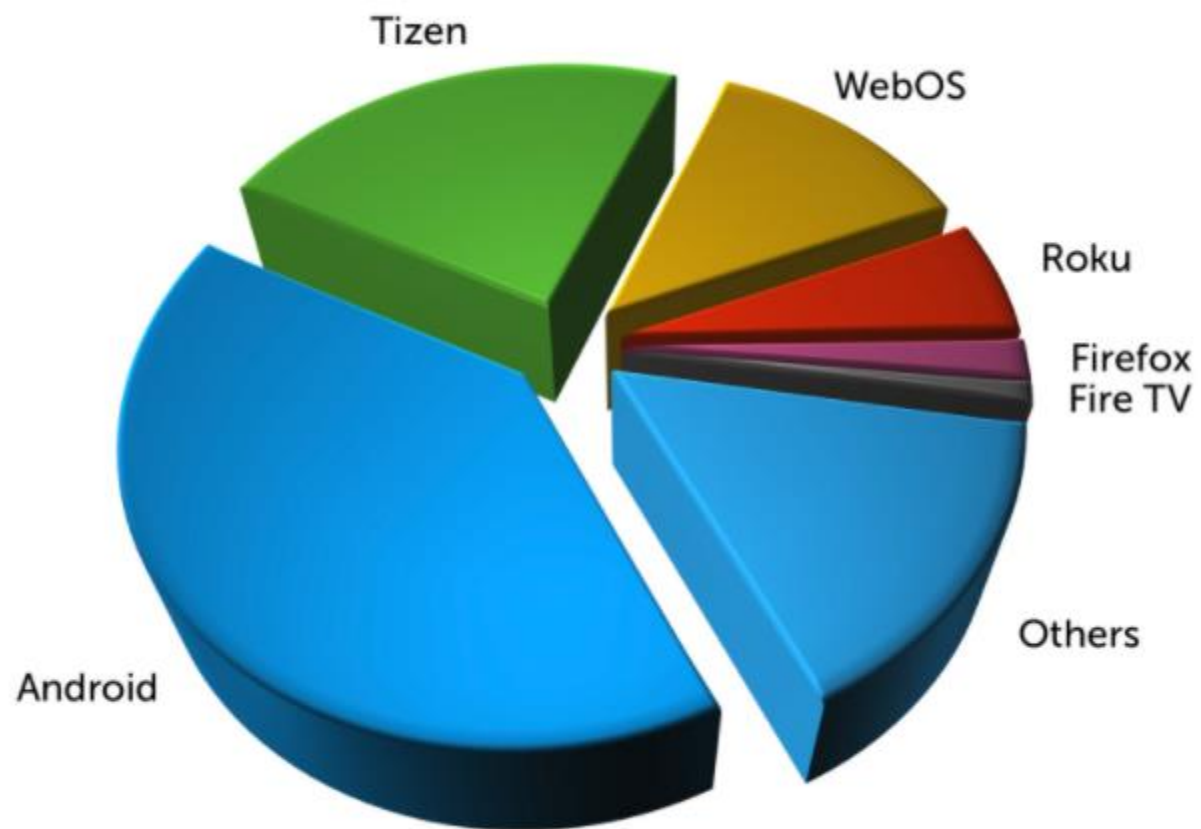
- Chromecast / Sabrina
- Android OS Sempre entre as “Top10”
- “Sete das dez principais marcas de TV do mundo usando o sistema do Google”!
- Existem 7000 Apps (Mobile: 3,1 Milhões)

# Motivação



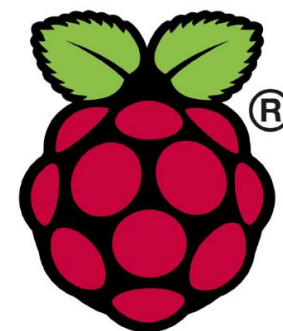


# Motivação



Global Smart TV OS Market Share per Wakefield Research. Image: Digitized House

# Motivação



**Raspberry Pi**

# Motivação mas com desafio!





# Possibilidade

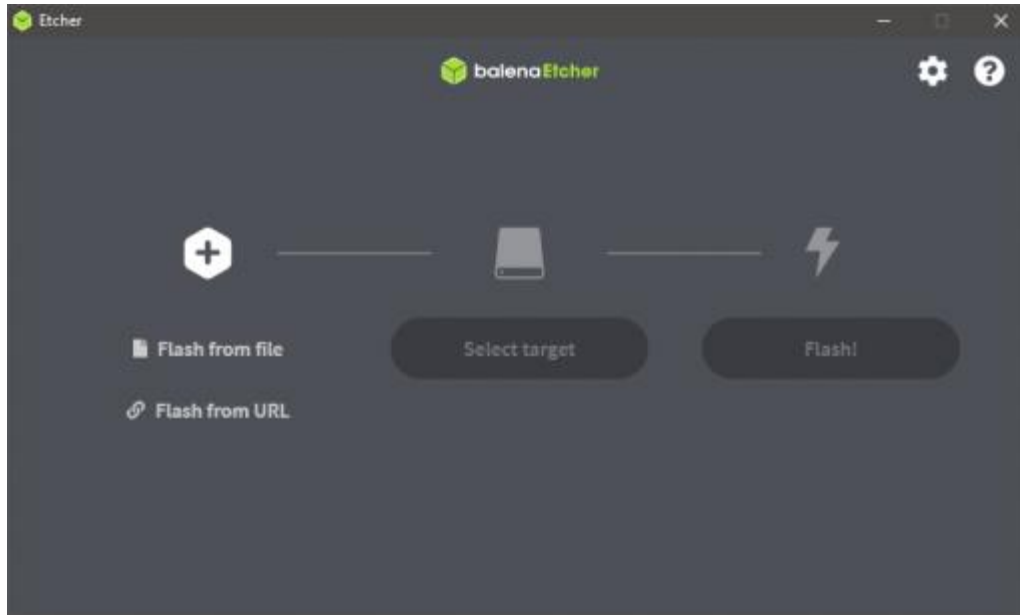


# Possibilidade

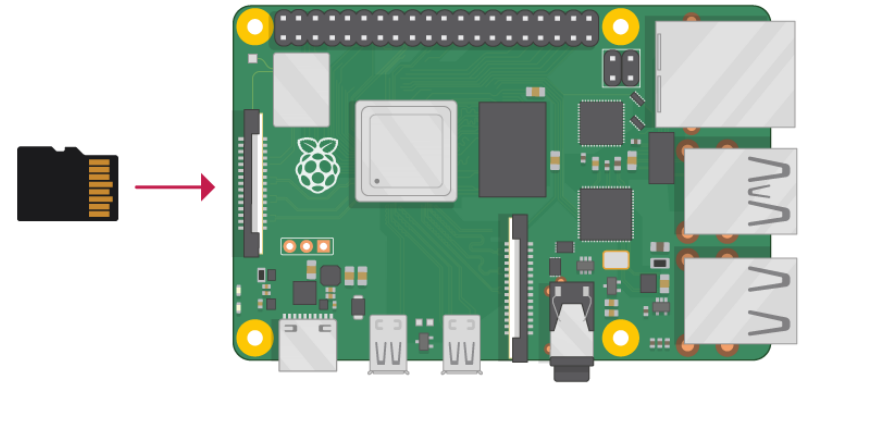


Lineage OS

Lineage OS Android Distribution



[etcher.download](https://etcher.download)





AVALIE A PALESTRA

# Componente TMediaPlayer



# Componente TMediaPlayer

Componente presente na VCL desde as primeiras versões do Delphi:



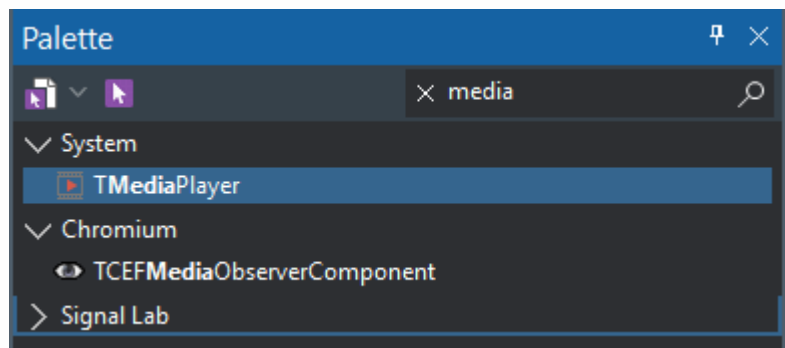
Utilizava a API do MCI (Media Control Interface) para controlar os dispositivos multimídia.

MCI ► Obsoleto com o advento da API do DirectX (1995)



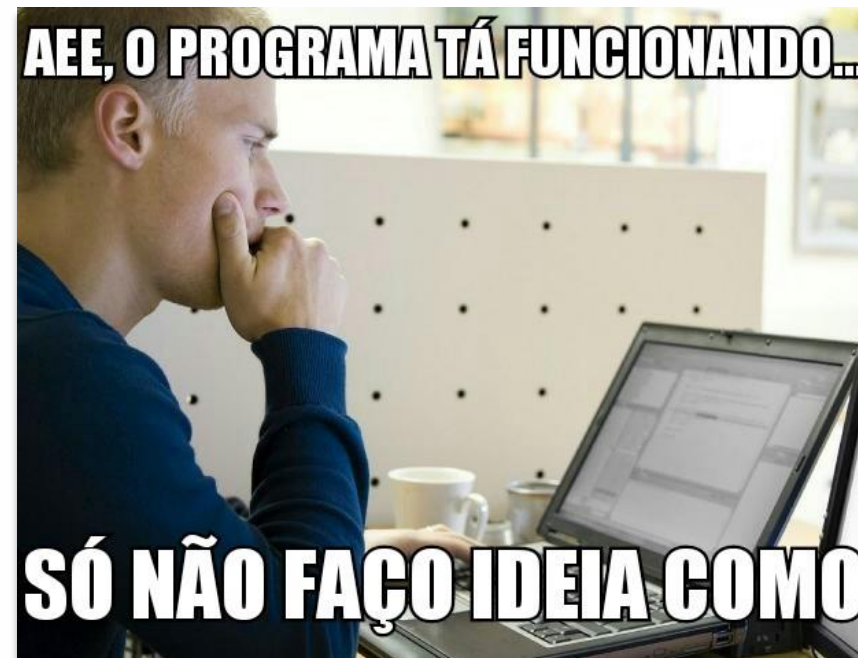
# Componente TMediaPlayer

E ele ainda existe na VCL!



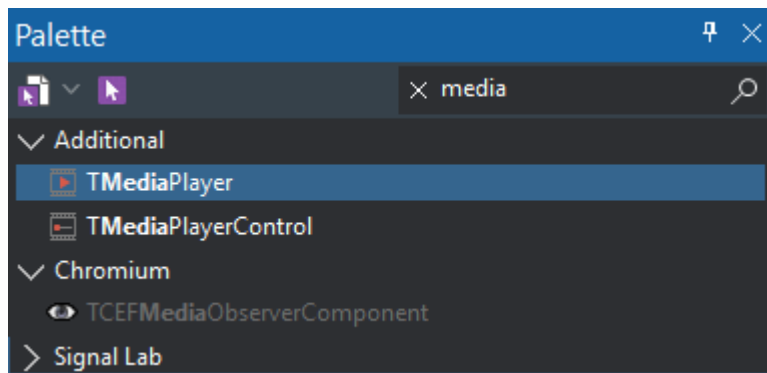
- Compatibilidade!

E provavelmente você não vai usar!



# Componente TMediaPlayer

E ele também existe na FMX!



Grande diferencial: Nativo e Multiplataforma!

E provavelmente você vai querer usar!



# Componente TMediaPlayer

Explorando o exemplo:

► FMX.StreamingMedia Sample

C:\Users\Public\Documents\Embarcadero\Studio\20.0\Samples\Object  
Pascal\Multi-Device Samples\Media\StreamingMedia

Porém, o Streaming do exemplo tem algumas limitações!





AVALIE A PALESTRA

# Adequações no FMX (Android)

# Componente TMediaPlayer (FMX)

Porém...

**FMX.Media.Andorid.pas initialization section:**

```
TMediaCodecManager.RegisterMediaCodecClass('.mov', SVMOVFiles, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.m4v', SVM4VFiles, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.mp4', SVMP4Files, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.3gp', SV3GPFiles, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.mp3', SVMP3Files, TMediaType.Audio, TAndroidMediaCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.caf', SVCAFFiles, TMediaType.Audio, TAndroidMediaCodec);
```

Foco em arquivos!



# Componente TMediaPlayer (FMX)

Mas o que precisamos para executar streaming de vídeo?

Executar links .M3U8:

M3U ou M3U8 (Media URL) é um formato que armazena listas de arquivos multimídia.

- Utilizado no protocolo HLS (HTTP Live Streaming)

# Componente TMediaPlayer (FMX)

Pequena modificação:

**FMX.Media.Andorid.pas initialization section:**

```
TMediaCodecManager.RegisterMediaCodecClass('.mov', SVMOVFiles, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.m4v', SVM4VFiles, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.mp4', SVM4Files, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.m3u8', SVMP4Files, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.3gp', SV3GPFiles, TMediaType.Video, TAndroidVideoCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.mp3', SVMP3Files, TMediaType.Audio, TAndroidMediaCodec);  
TMediaCodecManager.RegisterMediaCodecClass('.caf', SVCAFFiles, TMediaType.Audio, TAndroidMediaCodec);
```



Só que sem vídeo! ☹️

# Mudanças no FMX.Media.Android.pas

```

end;

procedure TAndroidVideo.CheckVideo;
const
  CYes = 'yes';
var
  MMR : JMediaMetadataRetriever;
  HasVideoValue: string;
begin
  if TOSVersion.Check(4, 0) and FileExists(FileName) then
  begin
    MMR := TJMediaMetadataRetriever.JavaClass.init;
    MMR.setDataSource(StringToJString(FileName));
    HasVideoValue := JStringToString(MMR.extractMetadata(TJMediaMetadataRetriever.JavaCl
    FVideoEnabled := CYes = HasVideoValue;
    MMR := nil;
  end
  else
    if (copy(FileName.ToUpper,1,4)='HTTP') then
      FVideoEnabled := True
    else
      FVideoEnabled := Control <> nil;
end;

procedure TAndroidVideo.RealignView;

procedure UpdateViewBounds;
var
  OriginalVideoRect: TRectF;
  FitRect: TRectF;
  HMargins: Single;
  VMargins: Single;
  FVideoEnabled := Control <> nil;
end;

```

```

end;

procedure TAndroidVideo.CheckVideo;
const
  CYes = 'yes';
var
  MMR : JMediaMetadataRetriever;
  HasVideoValue: string;
begin
  if TOSVersion.Check(4, 0) and FileExists(FileName) then
  begin
    MMR := TJMediaMetadataRetriever.JavaClass.init;
    MMR.setDataSource(StringToJString(FileName));
    HasVideoValue := JStringToString(MMR.extractMetadata(
    FVideoEnabled := CYes = HasVideoValue;
    MMR := nil;
  end
  else
    FVideoEnabled := Control <> nil;
end;

procedure TAndroidVideo.RealignView;

procedure UpdateViewBounds;
var
  OriginalVideoRect: TRectF;
  FitRect: TRectF;
  HMargins: Single;
  VMargins: Single;
  FVideoEnabled := Control <> nil;
end;

```



# Mudanças no FMX.Media.Android.pas

```
function TVideoPool.UIGetInstance: TVideoInstance;
begin
    if FReadyToUse.Count = 0 then
        CreateOneMoreInstance;
    Result := FReadyToUse.First;
    FReadyToUse.Remove(Result);
    FUsed.Add(Result);
end;

procedure TVideoPool.UIReturnInstance(const AInstance: TVideoInstance);
begin
    UIFreezeInstance(AInstance);
    FUsed.Remove(AInstance);
    FReadyToUse.Add(AInstance);
end;

initialization
    VideoPool := TVideoPool.Create;
    TMediaCodecManager.RegisterMediaCodecClass('.mov', SVMOVFiles, TMediaType.Video, TAndroid);
    TMediaCodecManager.RegisterMediaCodecClass('.m4v', SVM4VFiles, TMediaType.Video, TAndroid);
    TMediaCodecManager.RegisterMediaCodecClass('.mp4', SVMP4Files, TMediaType.Video, TAndroid);
    TMediaCodecManager.RegisterMediaCodecClass('.m3u8', SVMP4Files, TMediaType.Video, TAndroid);
    TMediaCodecManager.RegisterMediaCodecClass('.3gp', SV3GPPFiles, TMediaType.Video, TAndroid);
    TMediaCodecManager.RegisterMediaCodecClass('.mp3', SVMP3Files, TMediaType.Audio, TAndroid);
    TMediaCodecManager.RegisterMediaCodecClass('.caf', SVCAFFFiles, TMediaType.Audio, TAndroid);

    TMediaCodecManager.RegisterMediaCodecClass(SAllFilesExt, SDefault, TMediaType.Video, TAndroid);

finalization
    VideoPool.Free;
```

```
function TVideoPool.UIGetInstance: TVideoInstance;
begin
    if FReadyToUse.Count = 0 then
        CreateOneMoreInstance;
    Result := FReadyToUse.First;
    FReadyToUse.Remove(Result);
    FUsed.Add(Result);
end;

procedure TVideoPool.UIReturnInstance(const AInstance: TVideoInst
begin
    UIFreezeInstance(AInstance);
    FUsed.Remove(AInstance);
    FReadyToUse.Add(AInstance);
end;

initialization
    VideoPool := TVideoPool.Create;
    TMediaCodecManager.RegisterMediaCodecClass('.mov', SVMOVFiles,
    TMediaCodecManager.RegisterMediaCodecClass('.m4v', SVM4VFiles,
    TMediaCodecManager.RegisterMediaCodecClass('.mp4', SVMP4Files,
    TMediaCodecManager.RegisterMediaCodecClass('.3gp', SV3GPPFiles,
    TMediaCodecManager.RegisterMediaCodecClass('.mp3', SVMP3Files,
    TMediaCodecManager.RegisterMediaCodecClass('.caf', SVCAFFFiles,

    TMediaCodecManager.RegisterMediaCodecClass(SAllFilesExt, SDefau

finalization
    VideoPool.Free;
```

# Download da Lista de Execução do Google Drive

Componente Indy (TIdHttp):



IdHTTP1



Componente NetHTTPClient (TNetHTTPClient)



NetHTTPClient1

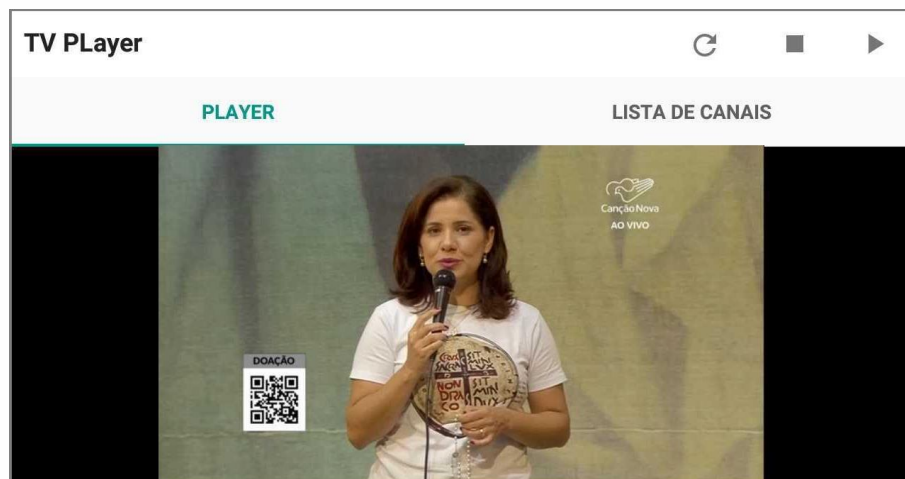
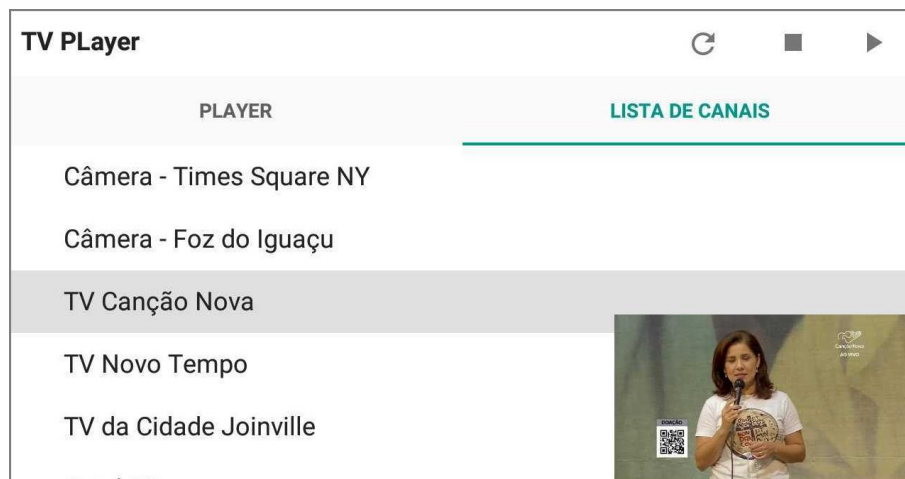


AVALIE A PALESTRA

# A Aplicação



# A Aplicação TV Player



# Vídeos de Demonstração

<https://www.youtube.com/channel/UC9ODw4kIS0Xjx0Fg-gyvR0Q/>



 **Jorge Manuel Lages Fernandes**  
35 subscribers

INÍCIO VÍDEOS LISTAS DE REPRODUÇÃO CANAIS DISCUSSÃO AC >

Carregamentos ► REPRODUZIR TODOS



**A ESTREAR AGORA**

Executando Aplicação de Vídeo com Firemonkey no...



**ESTREIA**

Executando Aplicação de Vídeo com Firemonkey no...

Estreia a 20/10/20, 09:39.

DEFINIR LEMBRETE



**D DEBUG - PARTE 2** 1:13:14

Universidade Delphi Desvendando o Debug do...

75 visualizações • há 4 meses



**DX DEBUG - PART** 1:39:05

Universidade Delphi Desvendando o Debug do...

214 visualizações • há 5 meses



AVALIE A PALESTRA

# Fazendo Funcionar!



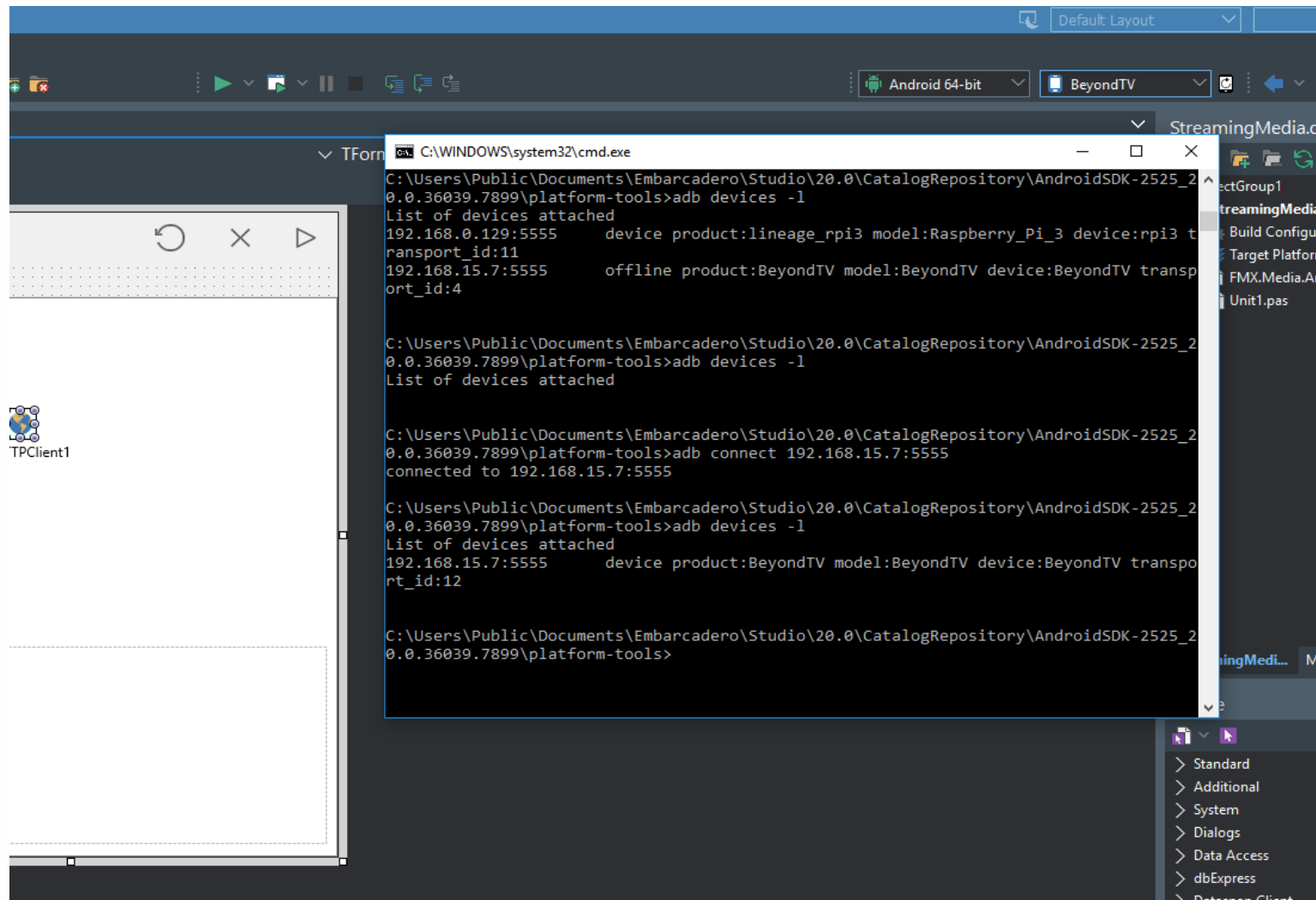
# Executando na Smart TV e no Raspberry

## Dicas Importantes:

- ADB via Wifi
  - `C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK-2525_20.0.36039.7899\platform-tools\adb.exe`  
connect IP:5555
- Desligando o Raspberry
  - Android: `reboot -p`
- Modo Desenvolvedor (Smart TV)



# Executando na Smart TV e no Raspberry



The screenshot shows the Android Studio IDE with a command prompt window open. The command prompt is running ADB commands to list and connect to devices. The background shows the IDE interface with a project named 'StreamingMedia.dp' and a file explorer on the right.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK-2525_2
0.0.36039.7899\platform-tools>adb devices -l
List of devices attached
192.168.0.129:5555    device product:lineage_rpi3 model:Raspberry_Pi_3 device:rpi3 t
ransport_id:11
192.168.15.7:5555    offline product:BeyondTV model:BeyondTV device:BeyondTV transp
ort_id:4

C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK-2525_2
0.0.36039.7899\platform-tools>adb devices -l
List of devices attached

C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK-2525_2
0.0.36039.7899\platform-tools>adb connect 192.168.15.7:5555
connected to 192.168.15.7:5555

C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK-2525_2
0.0.36039.7899\platform-tools>adb devices -l
List of devices attached
192.168.15.7:5555    device product:BeyondTV model:BeyondTV device:BeyondTV transpo
rt_id:12

C:\Users\Public\Documents\Embarcadero\Studio\20.0\CatalogRepository\AndroidSDK-2525_2
0.0.36039.7899\platform-tools>
```

# Desafios no Android da Smart TV

- Interação do Usuário
- Controle Remoto
- Deploy



<https://developer.android.com/training/tv/start/start>

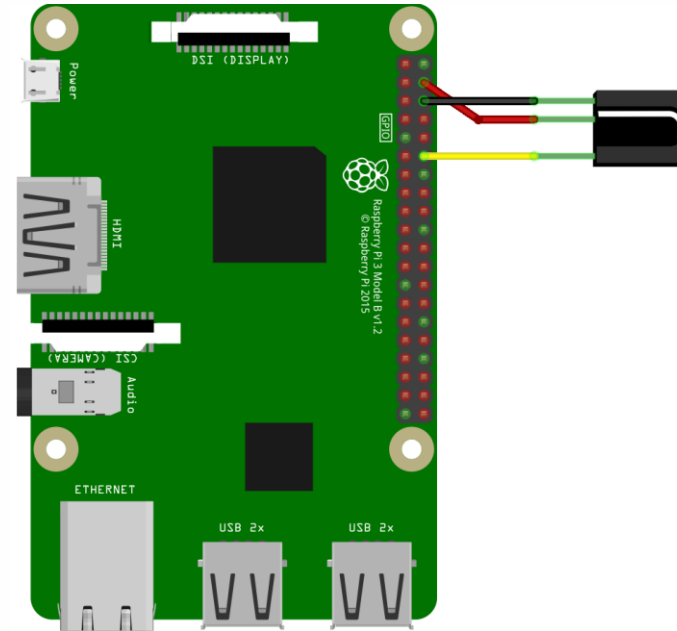


AVALIE A PALESTRA

# Próximos Passos

# Próximos Desenvolvimentos

- Melhorar a interação do usuário
- Adicionar outros formatos de streaming...
- Melhorar visual
- Integração ao “Controle Remoto” no Raspberry PI
- Fazer outras “brincadeiras” com o PI
- Testar no Linux!
- Apple TV?





## Considerações Finais

- Sim, é possível!
- É possível utilizar os recursos nativos do para multimídia!
- É possível utilizar muito do código fonte do FMX
- O Android é uma opção viável!
- Destaque para o Android TV
- Cuidado com o hardware (PI3)...
- O Delphi evoluiu, você também tem que evoluir!



- Jorge Manuel Lage Fernandes
- YoungTech Sistemas
- [jorge@youngarts.com.br](mailto:jorge@youngarts.com.br)
- [github.com/jmfjorge](https://github.com/jmfjorge)



AVALIE A PALESTRA

OBRIGADO

