**Android Auto**

Dia do lançamento: 19/03/201~5

**Sistema operativo**: Android 5.0+ (Lollipop) API level 21\*

**Objetivo**: Criar um sistema de Infotainment, que mantenha o utilizador informado sem ter de tocar no smartphone, tornando a condução mais segura. Todo o processamento é feito pelo smartphone, assim como a data usada. O smartphone fica inutilizável quando ligado ao Display.

**Carro:** Ligar telemóvel com Android\* através de cabo USB/microUSB a um carro com Dash compatível ou AfterMarket display compatível. Fig.1

**Telemóvel**: Telemóvel com Android\* e a App Android Auto (Apenas disponível em alguns países, Google PlayStore). Fig.2

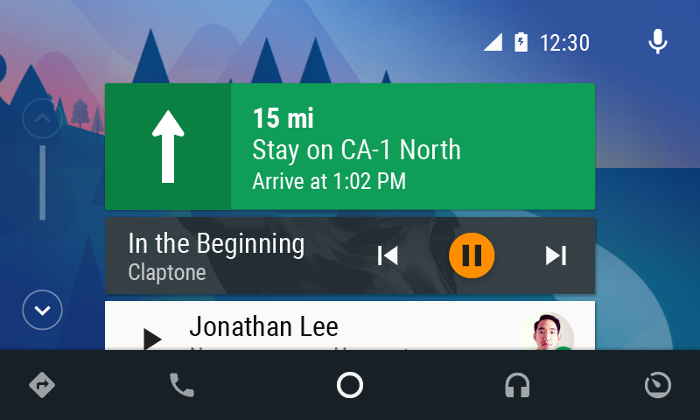
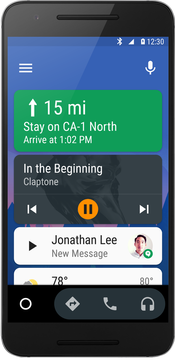
 

Figura 1: Android Auto no Dashboard Figura 2: Android Auto no smartphone

**Apps para Android Auto:**

- Audio apps: allow users to browse and play music and spoken audio content in the car.

- Messaging apps: receive incoming notifications, read messages aloud via text-to-speech, and send replies via voice input in the car.

**Criar Apps para Android Auto:**

**Pré-requisitos:**

* Criar projeto com API level 21 ou maior (Implementa Áudio playback e mensagens compatíveis com Android Auto. *Manifest targetSdkVersion > 21+.*
* Instalar a support library – para criar messaging apps para o Auto é preciso a Classe *NotificationCompat.CarExtender* (permite a criação de notificações compatíveis com dispositivo Auto).

**Declarar capacidades Auto:**

* Ficheiro de configuração XML Auto

Especificar as capacidades que a App vai usar no ficheiro XML (*res/xml/*). Exemplo: Audio app, criar *automotive\_app\_desc.xml* que deve ser guardado na pasta do projeto *res/xlm/ e deve conter a seguinte informação:*

*<automotiveApp>*

*<uses name="media" />*

*</automotiveApp>*

O elemento <uses> indica o componente Auto que a App pretende usar.

Values disponíveis para <uses> são:

* + media – app uses the Android framework APIs to play music in a vehicle
  + notification - app displays message notifications in the car’s Home screen, allows users to select messages to be read aloud, and lets them respond through voice input
* Add a manifest entry

No manifesto da app, inserir referencia a configuração XML Auto. Adicionar "com.google.android.gms.car.application" no elemento <application>.

<application  
    <meta-data android:name="com.google.android.gms.car.application"  
     android:resource="@xml/automotive\_app\_desc"/>  
  
</application>

**Auto app quality**

Auto UI deve ter como objetivo minimizar a distração do condutor.

Devem ser usados comandos por voz e um design visual simples. A aplicação deve prever e ser previsível, a informação deve ser apresentada apenas quando relevante e devem ser usados padrões simples e previsíveis para tarefas comuns. Se a aplicação não seguir todos os critérios para o Android Auto, a aplicação não é publicada.

* **Core app quality**

A aplicação deve seguir os critérios gerais de qualidade Android em adição aos critérios de qualidade relativos ao Android Auto.

*https://developer.android.com/develop/quality-guidelines/core-app-quality.html*

* **Visual design and user interaction**

Driver Attention, Layout, Visual Contrast, Interaction

* **Functionality**

General, Media, Notifications, Messaging

**Aplicações a funcionar com Android Auto**

**Audio Aplicações para ouvir musica, stream de podcasts, radio, etc.**

Google Play Music, Pandora, Spotify, Pocket Casts, Audible, Player FM, etc.

**Messaging Aplicações que leem mensagens de texto e permitem responder sem o utilizador ter de tocar no telemóvel.**

Hangouts, WhatsApp, Telegram, Text me, Kik, textplus, Skype

**Embedded Android (**Android Automotive**)**

**Modelos:** Audi Q8 and Volvo V90 SUVs

No Google I\O 17 foram apresentados carros da Audi e Volvo a correr o novo *Embedded Android for Automotive*.

Ao contrário do Android Auto, *Embedded Android* não necessita de nenhum smartphone conectado e mais importante, dá às marcas automóveis mais controlo sobre a aparência e que o sistema controla.

Apesar do core do Android ser igual a um smartphone/tablet, a fabricante automóvel pode decidir a user experience.

As vantagens para as fabricantes automóveis, é que os seus sistemas vão ter acesso a todas as aplicações Auto, podendo escolher quando e quais aplicações são utilizadas pelo utilizador.

No caso do Audi concept, existem 3 displays no carro ( central touchscreen, lower touchscreen onde se encontra o climate control e o Virtual Cockpit atras do volante). Enquanto os touchscreens principais correm Embedded Android, o Virtual Cockpit corre um costum real-time OS, isto por motivos de segurança de acordo com Google.

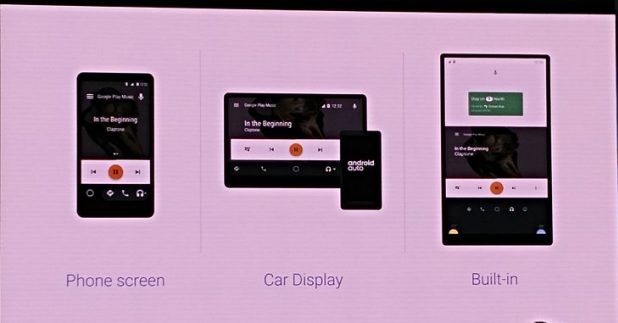
Apesar de serem OS diferentes, o Embedded Android passa informação para o Virtual Cockpit que é apresentada ao lado da velocidade, autonomia, etc.

https://www.slashgear.com/this-is-what-embedded-android-is-like-in-audis-google-io-concept-car-17485485/



Android Auto- aplicações definidas pela Google

Embedded Android- aplicações definidas pela fabricante automóvel



Unlike regular Android, Android Wear, and Android TV, Android Auto isn't an operating system. It doesn't live on the car's computer, it doesn't control peripherals, and it doesn't have an app store. Like Apple's CarPlay, Android Auto is just a "casted" interface. Your plugged-in smartphone sends a custom interface to the car's screen and receives touch events, but the car still has to run some other operating system.

Android Auto, which launched three years ago, was designed to provide drivers with an easy way to access information -- including navigation, streaming audio and communications -- while minimizing potential distractions.

Vantagens de Embedded Android:

* Android is based on Linux and has support for thousands of pieces of hardware
* Existem muitos apps developers de Android
* É open source
* No licencing fees for the kernel and Android OS
* Vantagem maior sao o numero de aplicações
* Google Maps e speech recognition da Google ja estao a ser desenvolvidos á anos.
* Third-party application. Android Auto esta limitado a apps de áudio e messaging
* Custom aplications de fabricantes de carros

Desvantagens:

* real-time capabilities
* big complex code bases - if you're developing a safety critical device that's always a bit of a concern
* security - Google has obviously had some struggles with that over the years with Android. They've made a lot of improvements and now it's based on security enhanced Linux and they've actually added some additional security features around the vehicle hardware abstraction layer in this new infotainment platform