

# Prácticas Fundamentos de Redes: Chromecast

Francisco Javier Morales Piqueras  
Rubén Morales Pérez

12 de diciembre de 2016

# Índice

## ① Introducción

## ② Software

- Modos de funcionamiento

- Arquitectura

## ③ Protocolos

- DIAL

- mDNS

- Comparación otros protocolos

- Miracast

## ④ Código de ejemplo de una aplicación

# Introducción

## Google Chromecast

Google Chromecast es un dispositivo de reproducción multimedia. Se conecta a una televisión o monitor vía HDMI y hace streaming mediante Wi-Fi. El contenido puede alojarse en un dispositivo conectado a una red local o en un servidor externo.

## Streaming

Utiliza el software propietario Google Cast, controlando la reproducción multimedia en un receptor desde uno o varios dispositivos locales. Es compatible con Android, iOS, Chrome OS y aplicaciones de Google Chrome.

# Tipos

## Vídeo y audio

- Chromecast primera generación
- Chromecast segunda generación (compatible Wi-Fi 5GHz)
- Chromecast Ultra (reproducción 4k)

## Audio

- Chromecast Audio

# Chromecast primera generación

## Hardware

- 512 MB de Micron DDR3L RAM
- 2/4 GB de memoria flash

Incluye salida HDMI, entrada micro USB para alimentación, un LED que indica el estado del dispositivo y un botón de reset.



# Chromecast segunda generación

## Diferencias

256 MB de memoria flash, procesador con dos núcleos y tres antenas para mejorar la conexión con el router.



# Chromecast Audio

## Diferencias

Salida MiniJack de 3,5mm en lugar del HDMI.



# Chromecast Ultra

## Diferencias

Entrada Ethernet para conexión a Internet.

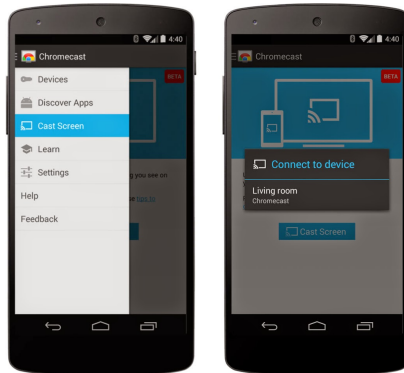




# Software

## Google Cast

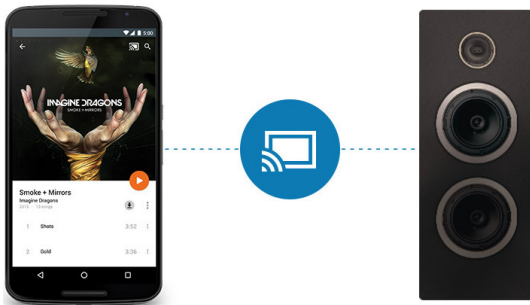
Google Chromecast es un dispositivo que actúa como receptor y es compatible con el protocolo propietario Google Cast. Para iniciar la reproducción de un contenido pulsamos el botón de *cast*.



# Funcionamiento

## Primer modo

Usar el dispositivo emisor para controlar la reproducción. El receptor (ej: Chromecast) se encarga de descargarlo del servidor, liberando al emisor de esta tarea. Esto permite al emisor ahorrar batería, estar bloqueado o en otra aplicación mientras la reproducción tiene lugar.



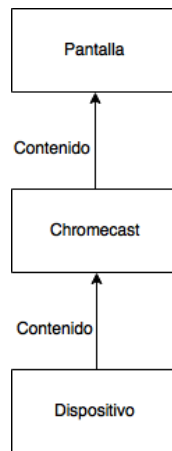
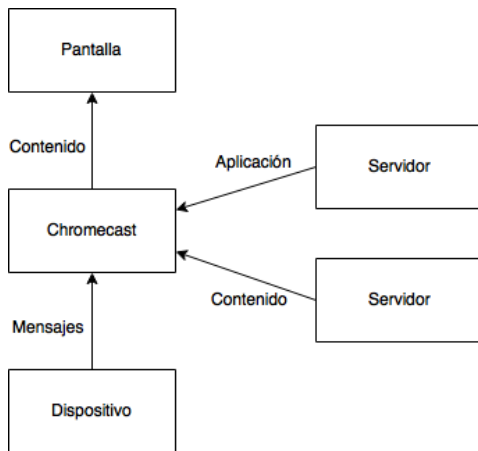
## Segundo modo

Diseñado para enviar contenido del emisor, como cuando hacemos mirroring o usamos la televisión como segunda pantalla.

La calidad del streaming en este caso varía según la potencia de procesamiento del emisor. En el caso de un smartphone la calidad de las imágenes normalmente se deteriora debido al escalado.



# Comparativa



# Arquitectura

## Google Cast

Google Cast implementa el paradigma del productor-consumidor.

## Aplicaciones

La aplicación emisora se encarga de controlar la reproducción y elegir el dispositivo donde se emite el contenido.

La aplicación receptora es una aplicación web ejecutándose en una adaptación de Chrome.

El código de la misma debe estar alojado en un servidor, ya que el Chromecast no almacena aplicaciones. Por tanto, aunque el contenido esté alojado en un dispositivo de la red local, seguirá necesitando conexión a internet para cargar la web app.

# Protocolos

## Estructura Google Cast

- Detección de dispositivos (mDNS)
- Intercambio de mensajes (Software propietario)

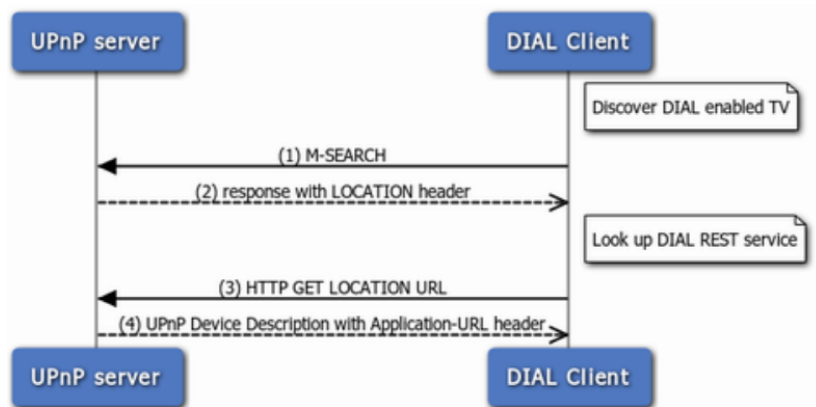
La primeras versiones del Chromecast usaban el protocolo DIAL que englobaba ambos pasos.

# DIAL

DIAL (Discovery And Launch) es el antiguo protocolo de comunicación. Se basa en Universal Plug and Play (UPnP), Simple Service Discovery Protocol (SSDP) y protocolos HTTP. SSDP sirve para la búsqueda de dispositivos UPnP. Utiliza UDP en unicast o multicast en el puerto 1900 para anunciar los servicios de un dispositivo.

El protocolo DIAL tiene dos componentes, DIAL Service Discovery y DIAL REST Service.

# Imagen





# mDNS

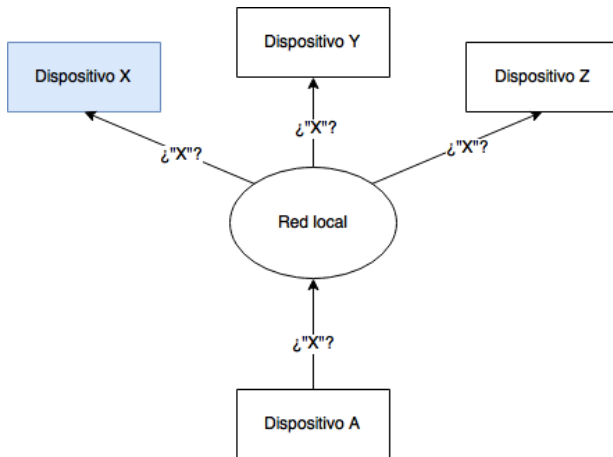
## mDNS

Multicast Domain Name System es la implementación del protocolo de resolución de nombres (DNS) para redes de área local, donde no existe un servidor DNS real.

## Ventajas

- Poca configuración para activarse
- Funciona cuando no hay infraestructura
- Soporta fallos en la infraestructura

# Primer paso mDNS



# Petición

## Ejemplo

```
00 00 00 00 00 01 00 00  00 00 00 00 07 61 70 70
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 00 01
```

- Flag de petición
- Nombre de dominio del servidor (appletv.local)

# Petición

## Ejemplo

```
00 00 00 00 00 01 00 00  00 00 00 00 07 61 70 70  
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 00 01
```

- Flag de **petición**
- Nombre de dominio del servidor (appletv.local)

# Petición

## Ejemplo

```
00 00 00 00 00 01 00 00  00 00 00 00 07 61 70 70  
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 00 01
```

- Flag de **petición**
- Nombre de dominio del servidor (**appletv.local**)

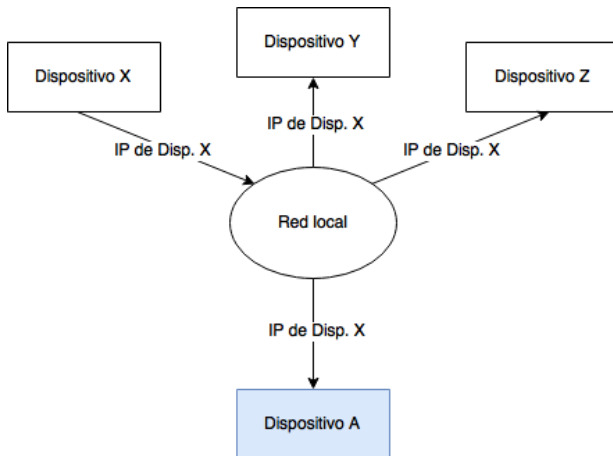
# Petición

## Ejemplo

```
00 00 00 00 00 01 00 00  00 00 00 00 07 61 70 70
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 00 01
```

- Flag de **petición**
- Nombre de dominio del servidor (appletv.**local**)

## Segundo paso mDNS



# Respuesta

## Ejemplo

```
00 00 84 00 00 00 00 01  00 00 00 02 07 61 70 70
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 80 01 00
00 78 00 00 04 99 6d 07  5a c0 0c 00 1c 80 01 00
00 78 00 00 10 fe 80 00  00 00 00 00 00 02 23 32
ff fe b1 21 52 c0 0c 00  2f 80 01 00 00 78 00 00
      08 c0 0c 00 04 40 00 00  08
```

- Flag de respuesta
- Bytes de dirección IPv4
- Bytes de dirección IPv6



# Respuesta

## Ejemplo

```
00 00 84 00 00 00 00 01  00 00 00 02 07 61 70 70
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 80 01 00
00 78 00 00 04 99 6d 07  5a c0 0c 00 1c 80 01 00
00 78 00 00 10 fe 80 00  00 00 00 00 00 02 23 32
ff fe b1 21 52 c0 0c 00  2f 80 01 00 00 78 00 00
      08 c0 0c 00 04 40 00 00  08
```

- Flag de **respuesta**
- Bytes de dirección IPv4
- Bytes de dirección IPv6

# Respuesta

## Ejemplo

```

00 00 84 00 00 00 00 01  00 00 00 02 07 61 70 70
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 80 01 00
00 78 00 00 04 99 6d 07  5a c0 0c 00 1c 80 01 00
00 78 00 00 10 fe 80 00  00 00 00 00 00 02 23 32
ff fe b1 21 52 c0 0c 00  2f 80 01 00 00 78 00 00
      08 c0 0c 00 04 40 00 00  08

```

- Flag de respuesta
- **Bytes de dirección IPv4**
- Bytes de dirección IPv6

# Respuesta

## Ejemplo

```

00 00 84 00 00 00 00 01  00 00 00 02 07 61 70 70
6c 65 74 76 05 6c 6f 63  61 6c 00 00 01 80 01 00
00 78 00 00 04 99 6d 07  5a c0 0c 00 1c 80 01 00
00 78 00 00 10 fe 80 00 00 00 00 02 23 32
ff fe b1 21 52 c0 0c 00  2f 80 01 00 00 78 00 00
08 c0 0c 00 04 40 00 00  08

```

- Flag de respuesta
- Bytes de dirección IPv4
- **Bytes de dirección IPv6**

# Otros procolos

## Miracast

Miracast es un protocolo multimedia para hacer streaming a un monitor desde un dispositivo local.

Con Miracast el dispositivo receptor es dependiente de que el dispositivo Android emisor se mantenga activo: si se bloquea también bloqueará la reproducción en el receptor.

## Capas

- Capa de internet: IPv4
- Capa de transporte: TCP/UDP
- Capa de aplicación, RTSP y RTP

## Red

La conexión está creada vía Wi-Fi Protected Setup (WPS), mecanismos para facilitar la configuración de una red WLAN con seguridad WPA2.

Existe una alternativa de código abierto a Miracast llamada [MiracleCast](#).

## Sin soporte de Google

A partir de Android 6.0, Google ha dejado de dar soporte nativo a Miracast en favor de su propio Google Cast.

# Ejemplo

## Enviar mensajes

Esta aplicación sirve para enviar texto desde una pestaña de Google Chrome y mostrarlo en una pantalla conectada a un Chromecast.

Esta aplicación consta de dos partes: la del emisor (Google Chrome) y la del receptor (Chromecast). Ambas son web apps en HTML que cargan un código JavaScript. Para ejecutarla debemos alojarla en un servidor.

# Creando servidor en el puerto 8000



A terminal window titled "CastHelloText-chrome-master — Python -m http.server — 127x30". The prompt shows the user is in the directory "MacBook-Pro-de-Francisco-Javier-Morales:CastHelloText-chrome-master". The command entered is "python3 -m http.server". The output is "Serving HTTP on 0.0.0.0 port 8000 ...".

```
MacBook-Pro-de-Francisco-Javier-Morales:CastHelloText-chrome-master fjmpiq$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 ...
```

# Aplicación emisora

## Chromehellotext.html

Desde Chrome, accedemos a la aplicación emisora (chromehellotext.html).



# Directorio del servidor



## Directory listing for /

---

- [.DS\\_Store](#)
  - [chromehello.txt.html](#)
  - [CONTRIBUTING.md](#)
  - [LICENSE](#)
  - [README.md](#)
  - [receiver.html](#)
-

# Aplicación emisora



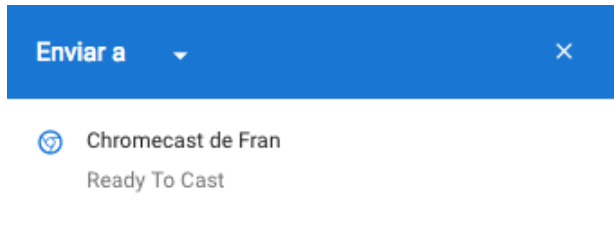
## Enviando contenido

A continuación, desde la extensión de Google Cast, seleccionamos enviar el contenido de la aplicación a nuestro Chromecast.

# Selección del Chromecast

## mDNS

Ahora es donde actúa mDNS



## Receiver.html

En ese momento, debería cargar en el Chromecast la aplicación receptora (receiver.html). Ya podemos escribir texto en el emisor y, al pulsar intro, debería aparecer en el televisor.



**Sample App**

**cosa**