

PRESENTACIÓN CARTOCIUDAD

LUZ FRIAS

12 MAY 2016

PRESENTACIÓN

QUÉ ES CARTOCIUDAD

- Proyecto del Instituto Geográfico Nacional
- Incorpora datos:
 - Catastro
 - Correos
 - INE
- Consulta de cartociudad
 - Web services abiertos
 - Visor

VISOR DE CARTOCIUDAD

- Visor
- Interesante!
 - Mapas base
 - Capas

VENTAJAS DE CARTOCIUDAD

- Sin límite de peticiones (como ocurre con Google APIs)
- Acceso a información administrativa:
 - Catastral
 - Sección censal (INE)

QUÉ ES CARTOCIUDAD

- Un paquete de R para facilitar el acceso a la información de las APIs
- Disponible en GitHub

```
library(devtools)  
install_github("cjgb/caRtociudad")
```

GEOCODING

- Devuelve las coordenadas de una dirección: completa

```
cartociudad_geocode("calle alameda 11, madrid")
```

```
##      priority status
## 1         1       1
## 2         2       2
## 3         3       2
##
##                                     comments
## 1                                     Portal/Pk encontrado.
## 2 Portal/Pk no encontrado. Se devuelve el más cercano encontrado
## 3 Portal/Pk no encontrado. Se devuelve el más cercano encontrado
##      province municipality road_type      road_name      road_fid
## 1      Madrid      Madrid      CALLE          ALAMEDA 280790007490
## 2      Madrid      Madrid      CALLE ALAMEDA DEL VALLE 280790010737
## 3      Madrid      Madrid      PASEO  ALAMEDA DE OSUNA 280790007489
##      numpk_name      zip      numpk_fid longitude latitude
## 1         11 28014 280790140461 -3.693777 40.41160
## 2         21 28051 280790150351 -3.622086 40.36707
## 3         7 28042 280790105955 -3.596491 40.45506
```

GEOCODING

- Devuelve las coordenadas de una dirección: a trozos

```
cartociudad_geocode(province = "valencia", municipality = "alzira",  
                    road_type = "calle",  
                    road_name = "verge de la murta",  
                    road_number = 11, zip = "46600")
```

```
##  priority status          comments          province  
## 1           1           1 Portal/Pk encontrado. Valencia/València  
##  municipality road_type          road_name          road_fid  
## 1           Alzira          CALLE VERGE DE LA MURTA 460170000252  
##  numpk_name   zip      numpk_fid  longitude latitude  
## 1           11 46600 460170006371 -0.4314195 39.15129
```


REVERSE GEOCODING

- Devuelve la dirección de unas coordenadas

```
cartociudad_reverse_geocode(40.4116, -3.69378)
```

```
##      tipo tipo.via nombre.via num.via  num.via.id municipio
## 1 portal   Calle   Alameda    11 280790140461  Madrid
##      provincia cod.postal
## 1      Madrid    28014
```

REVERSE GEOCODING

- ... pero cuidado con el fin de la tierra!

```
groenlandia <- c(lat = 60.289891, lon = -43.6352515)
resultado <- cartociudad_reverse_geocode(groenlandia['lat'],
                                          groenlandia['lon'])
resultado$provincia
```

```
## [1] "A Coruña"
```


MAPAS ESTÁTICOS



INDICACIONES

```
origen <- cartociudad_geocode("calle general perón 27, madrid")
destino <- cartociudad_geocode("calle alberto alcocer 46, madrid")

ruta <- get_cartociudad_route(c(origen$latitude, origen$longitudo),
                             c(destino$latitude, destino$longitudo),
                             vehicle = "walking")

ruta$distance
```

```
## [1] 1781.36
```

```
ruta$time
```

```
## [1] 213750
```

INDICACIONES

```
ruta$instructionsData[, c("description", "distance")]
```

```
##                               description
## 1          Continúe por AVENIDA GENERAL PERON
## 2      Gire a la izquierda por CALLE CAPITAN HAYA
## 3      Gire a la derecha por CALLE PEDRO TEIXEIRA
## 4          Gire a la izquierda por PASEO CASTELLANA
## 5      Gire justo a la derecha por CALLE GENERAL YAGÜE
## 6          Continúe por CALLE PROFESOR WAKSMAN
## 7 Gire ligeramente a la izquierda por CALLE PADRE DAMIAN
## 8          Gire a la derecha por AVENIDA ALBERTO ALCOCER
## 9                               Objetivo logrado
## distance
## 1         148
## 2         231
## 3         131
## 4         123
## 5          42
## 6         299
## 7         345
## 8         459
## 9           0
```

INFORMACIÓN ADMINISTRATIVA

- Devuelve datos de:
 - Reverse geocoding
 - Catastro
 - Sección censal
- Interesante para cruzar con otras fuentes (INE, ...)

INFORMACIÓN ADMINISTRATIVA

```
get_cartociudad_location_info(40.473219, -3.7227241,  
                               year = 2015,  
                               info.source = "census")
```

```
##          seccion distrito provincia municipio  
## 1 2807908148 2807908 Madrid Madrid
```


INFORMACIÓN ADMINISTRATIVA

```
get_cartociudad_location_info(40.473219, -3.7227241,  
                               year = 2015,  
                               info.source = "cadastre")
```

```
## $ref.catastral  
## [1] "8808803VK3880H"  
##  
## $url.ref.catastral  
## [1] "https://ww1.sedecatastro.gob.es/CYCBienInmueble/OVCListaBienes.aspx?del=28&mu"
```

DOCUMENTACIÓN DE LAS APIS

- Todas estas APIs están *documentadas*
- Esto es lo que nos imaginamos...



/countries

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

GET	/api/countries/	Gets a list of countries
-----	-----------------	--------------------------

POST	/api/countries/	Gets a list of countries
------	-----------------	--------------------------

GET	/api/countries/{pk}/	Detailed view of the country
-----	----------------------	------------------------------

PUT	/api/countries/{pk}/	Detailed view of the country
-----	----------------------	------------------------------

DELETE	/api/countries/{pk}/	Detailed view of the country
--------	----------------------	------------------------------

PATCH	/api/countries/{pk}/	Detailed view of the country
-------	----------------------	------------------------------

/manufacturers

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

/cigars

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

/custom

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

DOCUMENTACIÓN DE LAS APIS

- ... y **esta** la realidad

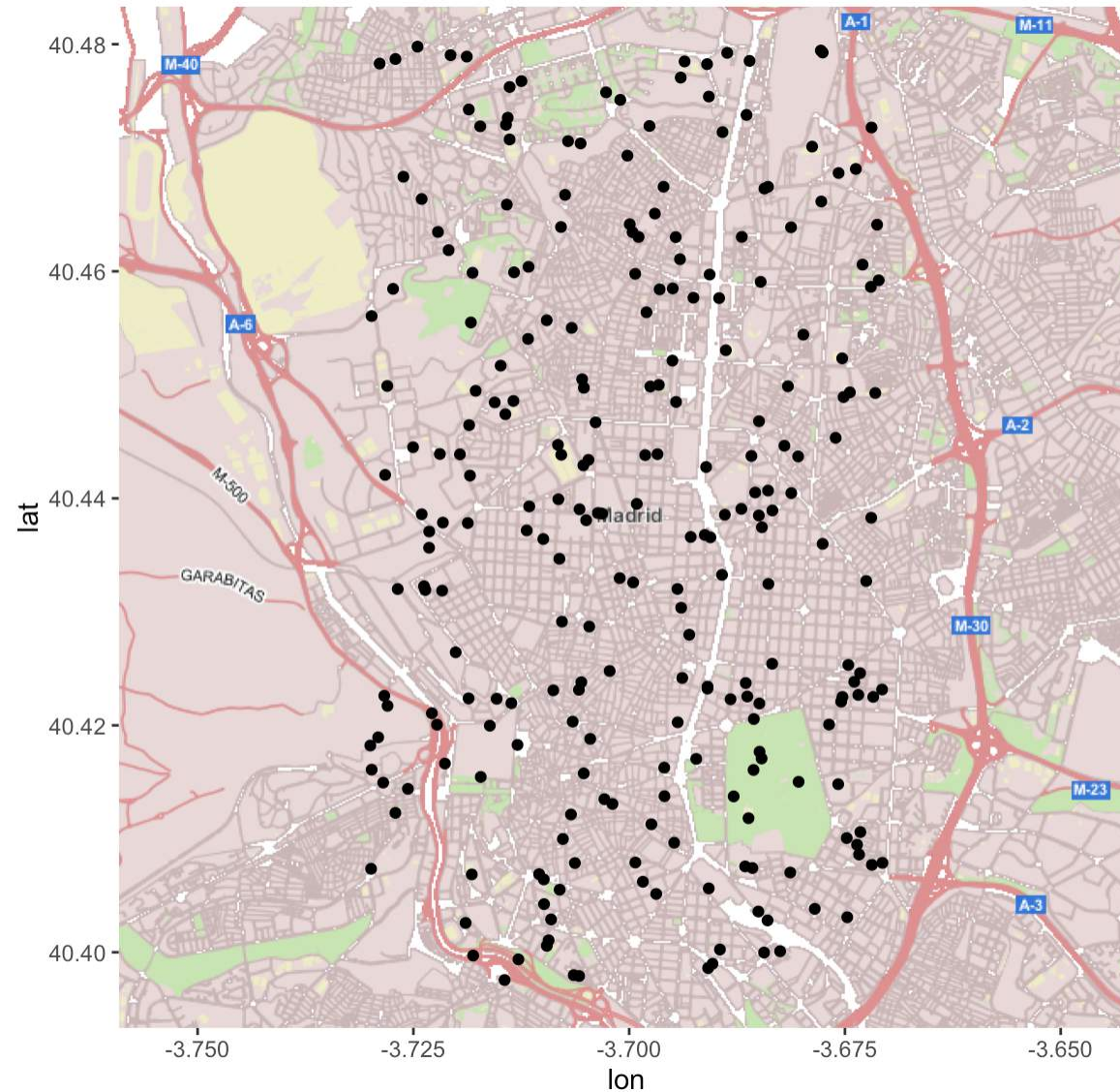
INGENIERÍA INVERSA

- Pero... ¿si ya estábamos haciendo la llamada!
- Aquí
- Podemos hacer ingeniería inversa con las herramientas del desarrollador

VISUALIZACIÓN

```
# Los pintamos  
centro <- c(so.lat + (ne.lat - so.lat)/2,  
           so.lon + (ne.lon - so.lon)/2)  
mapa <- get_cartocidadmap(centro, 5)  
ggmap(mapa) + geom_point(aes(x = lon, y = lat), data = localizaciones)
```

VISUALIZACIÓN



INFORMACIÓN ADMINISTRATIVA

```
registerDoParallel(cores = 4)
parsea.campo <- function(x) { ifelse(is.null(x), NA, x) }
parsea.respuesta <- function(res) {
  list(seccion          = res$seccion,
       ref.catastral    = parsea.campo(res$ref.catastral),
       url.ref.catastral = parsea.campo(res$url.ref.catastral),
       direccion        = paste(res$tipo.via, res$nombre.via, res$num.via))
}
secciones <- foreach(l=iter(localizaciones, by = "row"), .combine = rbind) %dopar% {
  tryCatch({
    parsea.respuesta(get_cartociudad_location_info(l$lat, l$lon, year = 2015))
  }, error = function(err) { # Si no hay información catastral salta un error
    parsea.respuesta(get_cartociudad_location_info(l$lat, l$lon, year = 2015,
                                                    info.source = c("census", "reverse")))
  })
}
localizaciones$seccion          <- unlist(secciones[, "seccion"])
localizaciones$ref.catastral    <- unlist(secciones[, "ref.catastral"])
localizaciones$url.ref.catastral <- unlist(secciones[, "url.ref.catastral"])
localizaciones$direccion        <- unlist(secciones[, "direccion"])
```

CRUZAMOS CON EL INE

```
# Cargamos la información del INE a nivel sección censal
sec.ine <- read.csv("dat/C2011_ccaa13_Indicadores.csv")

# Unificamos formato del identificador de sección
sec.ine$seccion <- paste0(sprintf("%02d", sec.ine$cpro),
                          sprintf("%03d", sec.ine$cmun),
                          sprintf("%02d", sec.ine$dist),
                          sprintf("%03d", sec.ine$secc))

# Nos quedamos con las columnas relativas al nivel educativo
sec.ine$pob.info.estudios <- sec.ine$t12_1 + sec.ine$t12_2 +
                             sec.ine$t12_3 + sec.ine$t12_4 + sec.ine$t12_5
sec.ine$pct.con.eso      <- (sec.ine$t12_4 + sec.ine$t12_5) /
                             sec.ine$pob.info.estudios
sec.ine$pct.est.grado.3  <- sec.ine$t12_5 / sec.ine$pob.info.estudios
columnas <- c("seccion", "pct.con.eso", "pct.est.grado.3")

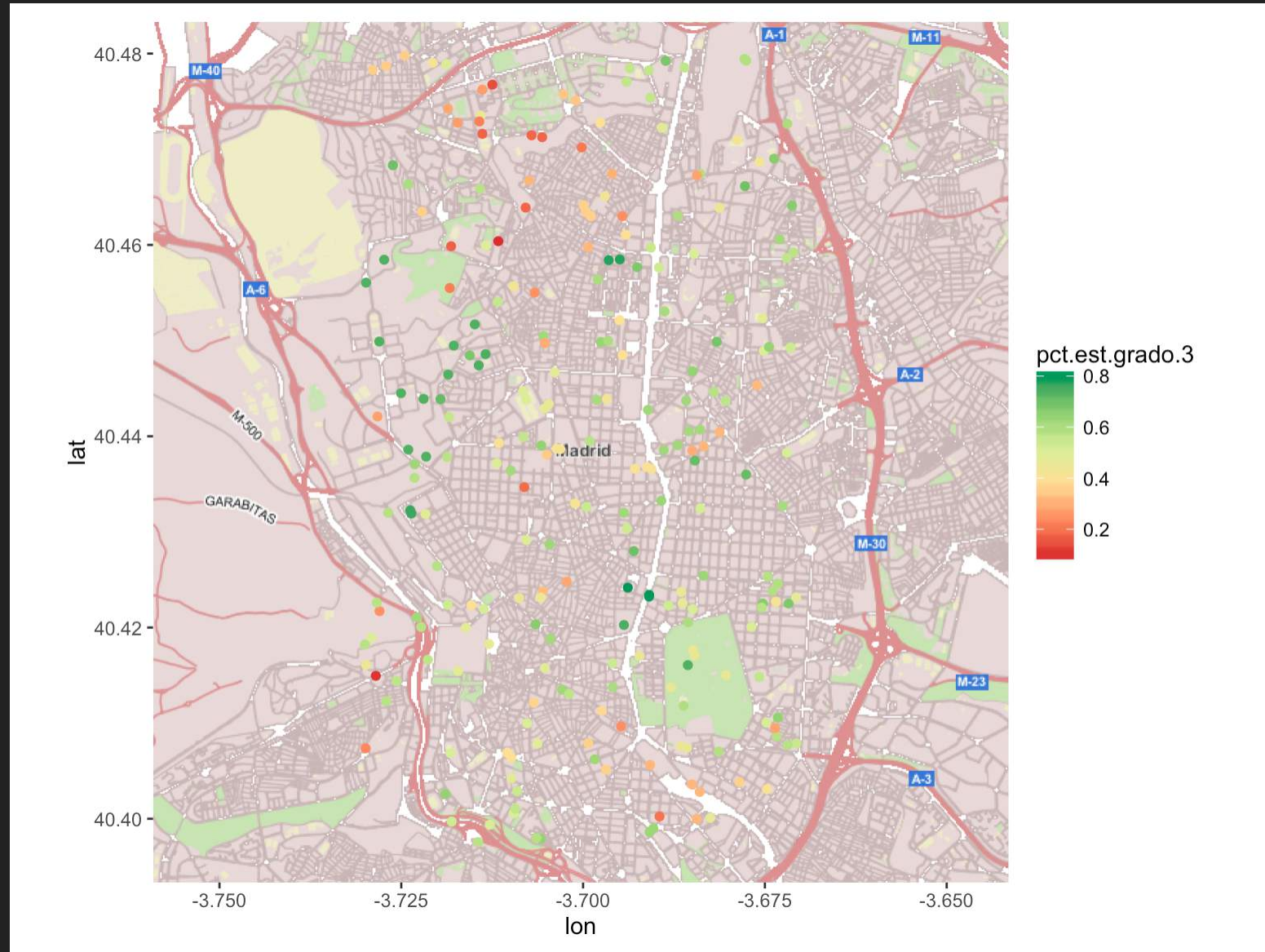
# Cruce con nuestros datos
locs <- merge(localizaciones, sec.ine[, columnas], by = "seccion")
```

MAPA ESTÁTICO

```
# Paletas de colores
pal.con.eso <- colorNumeric(
  palette = "RdYlGn",
  domain = locs$pct.con.eso
)
pal.grado.3 <- colorNumeric(
  palette = "RdYlGn",
  domain = locs$pct.est.grado.3
)
```

```
ggmap(mapa) +
  geom_point(aes(x = lon, y = lat, color = pct.est.grado.3), data = locs) +
  scale_color_distiller(palette = "RdYlGn", direction = 1)
```

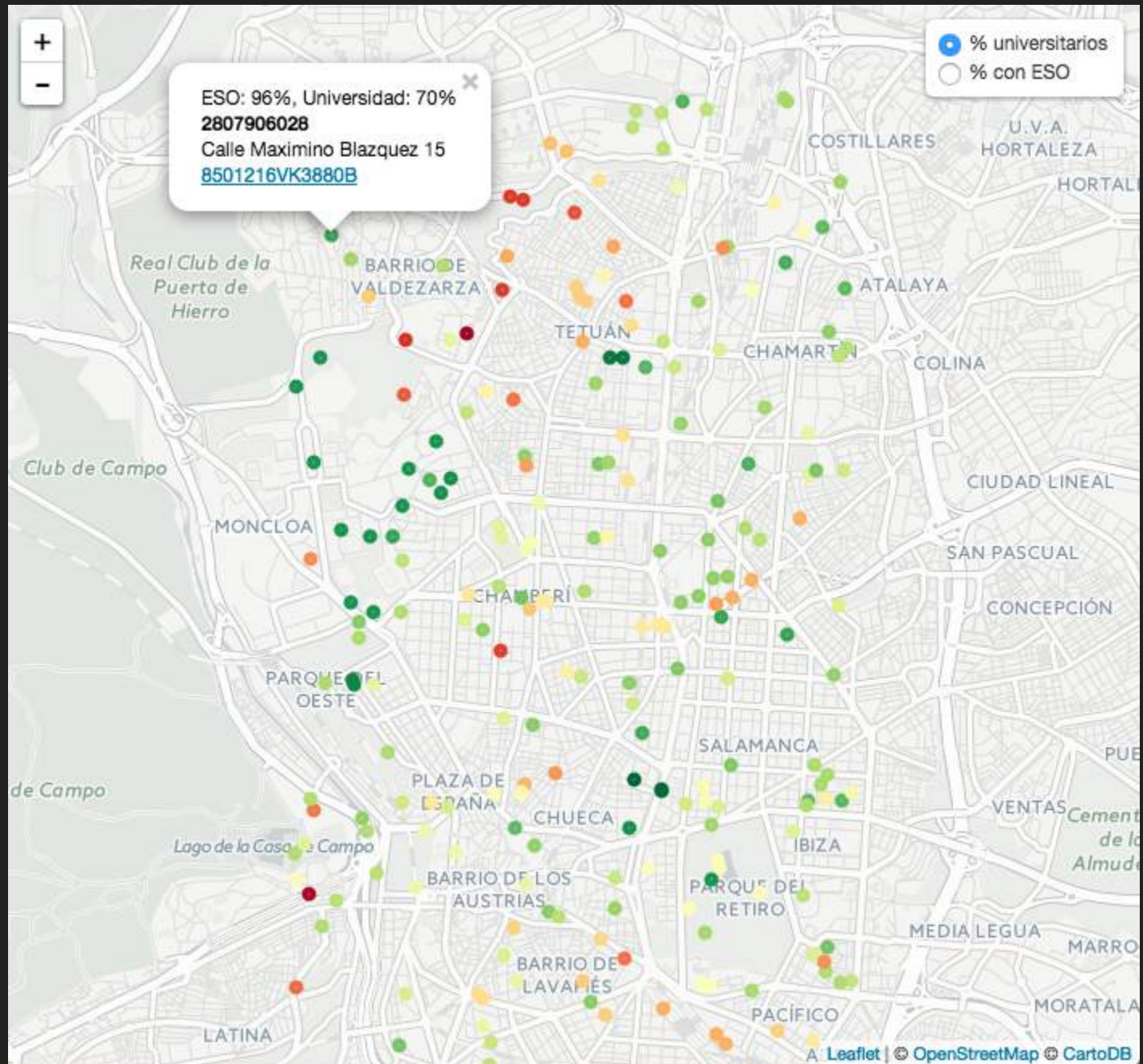
MAPA ESTÁTICO



MAPA DINÁMICO

- Resultado

```
# Popup
locs$popup <- paste0("<span>ESO: ", round(locs$pct.con.eso, 2) * 100,
                    "%, Universidad: ", round(locs$pct.est.grado.3, 2) * 100,
                    "%</span><br/>", "<b>", locs$seccion,"</b><br/>",
                    "<span>",locs$direccion,"</span>")
locs[!is.na(locs$ref.catastral),]$popup <-
  paste0(locs[!is.na(locs$ref.catastral),]$popup, "<br/><a href='",
         locs[!is.na(locs$ref.catastral),]$url.ref.catastral,"'>",
         locs[!is.na(locs$ref.catastral),]$ref.catastral, "</a>")
# Pintamos
map <- leaflet(locs) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addCircleMarkers(radius = 2, opacity = 1, lng = ~lon, lat = ~lat, popup = ~ popup,
                  color = ~pal.grado.3(pct.est.grado.3), group = "% universitarios")
  addCircleMarkers(radius = 2, opacity = 1, lng = ~lon, lat = ~lat, popup = ~ popup,
                  color = ~pal.con.eso(pct.con.eso), group = "% con ESO") %>%
  addLayersControl(
    baseGroups = c("% universitarios", "% con ESO"),
    options = layersControlOptions(collapsed = FALSE)
  )
```



¡GRACIAS!

- ¿Preguntas?