Proceso de sistematización (Mapeo)

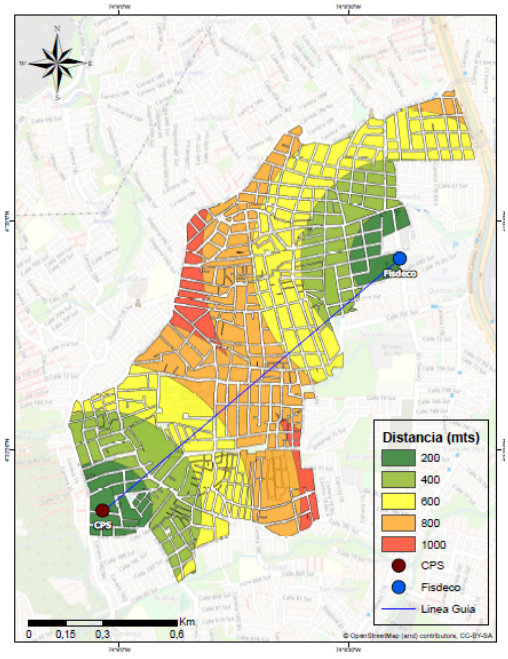
septiembre 2020

Segunda etapa del Mapeo.

La segunda parte del mapeo se utilizan softwares geográficos como ArcGIS y Google Earth Pro para adquirir la información de los establecimientos comerciales, ArcGIS Survey 123 para organizar la base de datos y el lenguaje R para analizar los datos.

Área de estudio

El objetivo de la georreferenciación es adquirir datos cualitativos de los establecimientos en la zona de CPS y FISDECO. Para ello se adquieren los shapefiles de los barrios y las manzanas de Bogotá D.C para encontrar el área de estudio.



PROXIMIDAD DE CPS Y FISDECO

Author: Proyecto REPILO Date: 13/07/2020
Coordinate System: GCS WGS 1984

Dicha área deber tener una conexión entre CPS y FISCECO y los establecimientos deben estar a una distancia de máximo 800 mts de CPS y FISCECO.
La zona de estudio encontrada comprende los barrios Bellavista Lucero Alto, Cedritos del Sur,La Estrella del Sur", Lucero del Sur y Naciones Unidas, Por lo tanto, se utilizó el programa ArcMap para encontrar el Área de estudio utilizando siguientes herramientas.

1. **Intersect:** Se delimita la zona de estudio con los barrios seleccionados y las manzanas correspondientes.
2. **Buffer:** Se delimita el área de estudio teniendo en cuenta una distancia de 200,400,600 y 800 mts de CPS y FISDECO.
3. **Polyline:** Se traza una línea desde CPS hasta FISDECO.

Teniendo como resultado el `Mapa 1`.

Variable	Clase	Definicion
Fid_barrio	numeric	Identificador de los Barrios
Scacodigo	factor	Código de los Barrios
Scanombre	factor	Nombre del barrio
Shape_leng	numeric	Longitud de la forma
Shape_area	numeric	Longitud del área
Fid_manz	numeric	Indentíficador de manzanas
Mancodigo	factor	Código de manzanas
X	numeric	Longitud
Y	numeric	Latitud

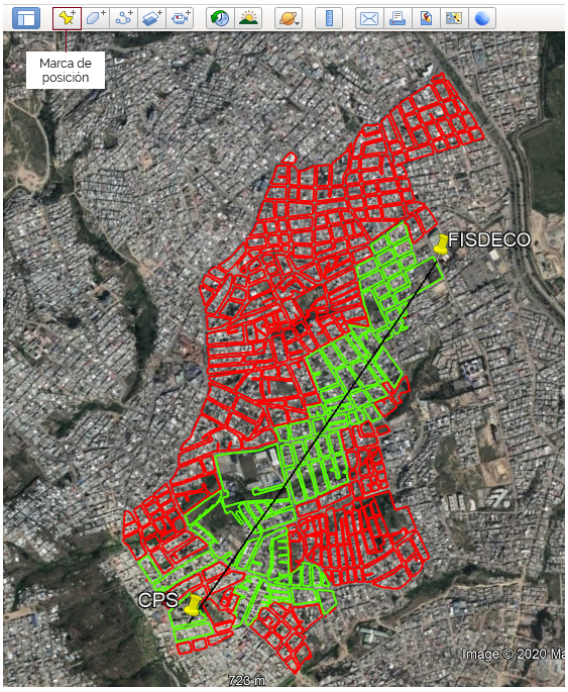
En la `Tabla 1` llamada `area_estudio`,se evidencian las variables extraídas del `Mapa 1` las cuales fueron utilizadas para la construcción de la base de datos.

Base de datos (`baserepilo`)

Para obtener los datos de los establecimientos comerciales se utilizaron dos programas ArcGIS Survey 123 y Google Earth Pro. Con Survey 123 se realizó un formulario para recopilar los datos de las siguientes variables.

Variable	Resultado
Fecha de Registro	Date
Código del encuestador	Numeric
Zona	CPS o Fisdeco
Código del establecimiento	Fid_manz_#
Tipo de actividad económica	Industrial,Comercial,Servicios,Financiero
Nombre del establecimiento	Character
Palabra Clave	Palabraclave.csv
¿El establecimiento tiene letrero?	Si,No
¿El establecimiento tiene letrero pintado?	Si,No
Celular	Numeric
Celular Secundario	Numeric
Teléfono	Numeric
Correo Electrónico	Character
Coordenadas	Longitud y Latitud
Fotografía	FID_manz_#.png

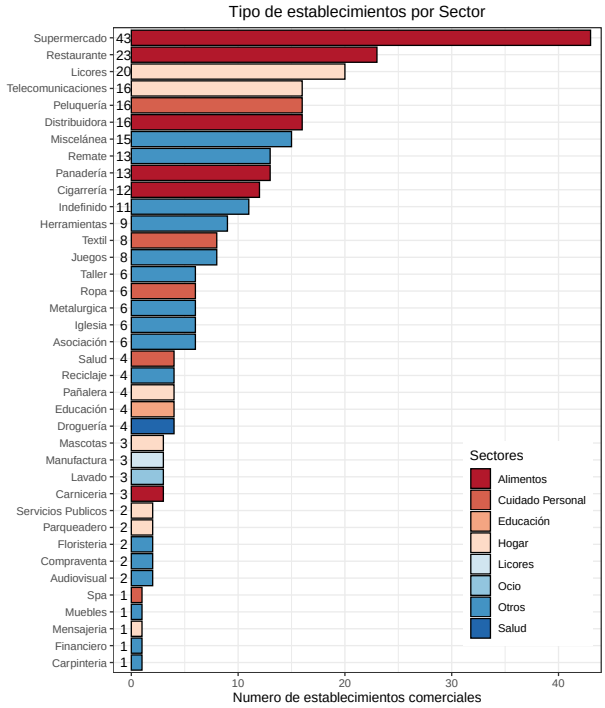
Como llave primaria se tuvo en cuenta la variable `Código del establecimiento` sujeta a la variable `Fid_manz` de la `Tabla 1` añadiendo una barra baja(_) y un número (#). Dicha variable también será utilizada para guardar las fotografías en la variable `fotografía`. Una vez realizado el formulario se comienza a recopilar los datos con `Google Earth Pro` siguiendo los siguientes pasos.



1. Exportar el archivo shapefile de la base `area_estudio` a formato KMZ en ArcMap con la herramienta Layer to KML (**link archivos kmz**).
2. Abrir el archivo KML en Google Earth Pro.
3. Crear una carpeta para almacenar los datos.
4. Abrir la vista Street View de Google Earth Pro (las imagenes son de Agosto de 2019).
5. Agregar marca de posición de los establecimientos y cambiar el color de la manzana de rojo a verde una vez complete todos los establecimientos comerciales de dicha manzana(**imagen N**)
6. Guardar la imagen del establecimiento con el `Código del establecimiento` como nombre.
7. Agregar el `Código del establecimiento` a la marca de posición en Google Earth Pro.
8. Insertar el `Código del establecimiento`, longitud y latitud de la marca de posición al formulario de Survey 123.
9. Insertar las demás variables cualitativas al formulario de Survey 123.

Resultados Generales

Como resultado de esta visita virtual, se identificaron 300 establecimientos los cuales se encuentran se organizaron por categorías. A continuación, se presenta un gráfico con la distribución por tipo de negocio, en los cuales se puede observar que la mayor cantidad de negocios está relacionada con supermercados, restaurantes, distribuidoras y licores.



De igual manera, la basepilo arrojo las siguientes características de los establecimientos comerciales.

- El 70% de los establecimientos tienen un letrero y el 30% no tiene letrero.
- El 67.67% de los establecimientos tienen nombre frente a un 32.33% de establecimientos sin nombre.
- El 96.44% de los locales no posee un medio de comunicación (Numero de teléfono, celular o correo electrónico) frente un 3.56% .
- Algunos de los letreros se encuentran deteriorados o casi borrosos.
- Gran parte de los negocios ubicados en la zona están relacionados con alimentos.
- En las calles donde hay mayor presencia de comerciantes son las calles cercanas a las vías principales de transporte o cercanas a los paraderos del Transmilenio.

Al dividir la investigación entre CPS y FISDECO durante la recolección de información se pudo observar aglomeraciones de establecimientos comerciales en las dos zonas. Por tal motivo, se realizó un clúster para determinar ¿en qué áreas se encuentran concentrados los establecimientos? y ¿cuántos establecimientos se encuentran en dichas áreas? utilizando la libreria leaflet de javaScript. La cual permite agrupar puntos en un mapa basados en su escala (realizando zoom al mapa). El mapa (N) se encuentra a una escala 3 y se evidencia que 140 establecimientos se encuentran cerca de CPS, 114 entre CPS y FISDECO y 46 en FISDECO, para un total de 150 establecimientos en las dos zonas (escala 2) y 300 establecimientos para la investigación final.

Los resultados generales que nos arroja la base de datos afirman que los establecimientos de la muestra responden a una demanda de primera necesidad como la alimentación, seguido la venta de licores por encima de bienes de secunda necesidad como las prendas de vestir. Además se corrobora que la interacción entre oferente y demandante es en su mayoría física debido a falta de uso de los medios de comunicación electrónicos.

Para obtener resultados específicos de la `baserepilo` se creó un dashboard que permite obtener información de cada uno de los 37 tipos de establecimientos con respecto a las demás variables.

Visualización de datos.

Despues de construir la base de datos en Survey 123 se extrae la información en formato `.csv` y se utilizá el lenguaje R para análizar los datos y realizar la visualización todo reunido en un dashboard hecho en shiny apps ver dashboard.

Descargar Base de datos.

La base de datos se encuentra subida a un repositorio en Github llamado REPILOS, la cual se puede descargar de tres formas.

1. Descargar en formato ZIP dando click en el botón Clone Download ZIP
2. Clonar el directorio con Git bash

```
$ git clone https://github.com/brianamaya01/REPILOS.git
```

3. Descargar desde RStudio

```
# 1. Elegir el directorio del trabajo con setwd
# 2. Crear una carpeta llamada REPILOS
if (!file.exists("REPILOS")){
    dir.create("REPILOS")
}

#3. Descargar el archivo .zip desde Github
download.file("https://github.com/brianamaya01/REPILOS/archive/master.zip",
              destfile = "REPILOS/REPILOS.zip", mode = "wb")
```

```
#4. Extraer archivo .Zip
unzip("REPILOS/REPILOS.zip",exdir = "./REPILOS")
```

Dashboard.

Objetivos

Una vez descargado la base de datos se procede a realizar un estudio de los posibles resultados que se pueden obtener con la base repilo. Llegando como resultado a los siguientes entregables, visualizados en un dashboard.

1. Un mapa con la geolocalización de los establecimientos comerciales determinado por `Tipo de establecimiento`.
2. Un clúster de los establecimientos comerciales basado en el `Tipo de establecimiento`.
3. Visualización de los establecimientos con sus respectivos datos.
4. Un gráfico de barras que muestre la actividad económica, el letrero, el tipo de letrero y el tipo de entrada de los establecimientos comerciales vasado en el `Tipo de establecimiento` general y por Zona.
5. Tres gráficos torta que muestren el porcentaje del tipo de letrero y el tipo de entrada y el letrero por `Tipo de establecimiento`.
6. Visualización de la base de datos con opción de descargarla.

Procedimiento

Para realizar el dashboard se debe instalar el paquete flexdashboard y abrir un documento flexdashboard para seguir con los siguientes pasos.

1 Añadir el directorio de trabajo: Se recomienda utilizar la carpeta `REPILOS` creada en el punto 4.1.

```
setwd("./REPILOS") #Añada la dirección correspondiente.
```

2 Cargar librerías: Después de cargan las librerías requeridas (para más información sobre las librerías escriba en la consola un signo de pregunta cerrado (?) seguido del nombre de la librerías.

```
library(tidyverse)
library(readxl)
library(leaflet)
library(sf)
library(htmltools)
library(plotly)
library(DT)
library(RColorBrewer)
library(scales)
```

3 Cargar bases de datos

```
baserepilo <- read_excel("baserepilo1.xlsx")
names(baserepilo) <- names(baserepilo) %>%
        str_to_title()
load("shapeFile.RData")# el archivo shapeFile.RData se guarda R como area_estudio
```

4 Primera pagina (georreferenciación)

4.1 Listas despegables: Se crean dos listas despegables, una llamada `palabra_clave` la cual posee el tipo de establecimiento comercial guardados en la variable `palabra_clave1` y otra llamada `image1` la cual posee el código del establecimiento en la variable `n_imagen`.

```
selectInput("palabra_clave","Tipo de Establecimiento",palabra_clave1[1])
selectInput("image1","Imagen:",n_imagen$Cod.establecimiento)
```

```
palabra_clave1 <- baserepilo%>%
        count(Palabra.clave)%>%
        arrange(desc(n))
n_imagen<- baserepilo %>%
        select(Cod.establecimiento) %>%
        mutate(orden = str_remove_all(Cod.establecimiento,"_")) %>%
        arrange(orden)
```

4.2 Botones: Para lograr los objetivos 1 y 2 se crean dos botones que cambian la metodología para visualizar el mapa, un método normal que muestra los establecimientos y un método clúster que muestra el agrupamiento espacial de los establecimientos.

```
radioButtons("metodo", "Metodo", c("Normal" = "normal","Cluster" = "cluster"))
```

4.3 Mapa: Se crea el mapa basado en la lista despegable `palabra_clave` y en los botones `normal` y `cluster`.

```
leafletOutput("leaflet_map", width = "100%", height = "600") # Salida del mapa
output$leaflet_map<- renderLeaflet({
pal = colorFactor(palette = "Paired", domain = baserepilo$Palabra.clave)
#Color de los establecimientos comerciales

base <- baserepilo %>%
        filter(Palabra.clave == input$palabra_clave)
# Condicionar la base con la lista llamada palabra_clave

labels_map <- paste("<p>", "Actividad Económica:",base$Actividad.economica, "</p>",
                "<p>", "Nombre: ",base$N.establecimiento,"</p>",
                "<p>", "código: ",base$Cod.establecimiento,"</p>",
                "<p>", "Zona: ",base$Zona,"</p>"
                ) # Código Html para mostrar en el mapa los metadatos
Atributos <- leaflet() %>%
        addTiles() %>%
        addProviderTiles(provider = "CartoDB.DarkMatterNoLabels") %>% # mapa base
        addPolygons(data = area_estudio$geometry,
                    weight = 1,
                    smoothFactor = 0.5,
                    color = "white",
                    fillOpacity = 0.05)
# Se plotea los polígonos de la base area_estudio

switch(input$metodo,
       normal = Atributos <- addCircleMarkers(Atributos,data = base,
                                  lng = base$X,
                                  lat = base$Y,
                                  color = pal(input$palabra_clave),
                                  label = lapply(labels_map,HTML)),
       # Se visualiza el metodo normal
       cluster = Atributos <- addCircleMarkers(Atributos,data = base,
                           lng = base$X,
                           lat = base$Y,
                           clusterOptions = markerClusterOptions()))
        # Se visualiza el metodo cluster
print(Atributos)
})
```

4.4 imágenes: Para mostrar las imágenes se utiliza la lista despegable llamada `image1` y las imágenes recopiladas en Survey 123 guardadas en Github (ver carpeta).

```
uiOutput("image") #Salida de las imágenes

output$image<- renderUI({
        prueba1 <-baserepilo %>%
                filter(Cod.establecimiento == input$image1) %>%
                #Se condiciona la base prueba 1 con la lista llamada imagen1
                transmute(
                Fotos = str_c("https://github.com/brianamaya01/REPILOS/blob/master/Fotos/"
                        ,Cod.establecimiento,".PNG?raw=true"))
                #Se crea una variable llamada fotos que contiene todas las direcciones
                #de las Fotos en Github
        tags$img(src=prueba1$Fotos, width = 400, height = 350)
        #Cogigo html que visualiza las fotos
        })
```

4.5 Tabla de variables: Se crea un tabla que contenga la respectiva información de cada imagen.

```
DT::renderDataTable({
        data<- baserepilo %>%
                select(Cod.establecimiento,N.establecimiento,
               Palabra.clave, Actividad.economica, Zona) %>%
                filter(Cod.establecimiento== input$image1) %>%
                gather(Variables, Resultado, N.establecimiento,
               Palabra.clave, Actividad.economica, Zona) %>%
                select(-Cod.establecimiento)
        #Se organiza la base para correr las variables correspondientes condicionada con
        #la lista llamada image1
        names(data) <- c("","")
  DT::datatable(data,
    rownames = FALSE,options = list(
    bPaginate = FALSE,dom = 'ltipr'
  )) # se pasa la base data a una datatable
})
```

5 Segunda Pagina (Grafícas)

5.1 Base de datos: Se crea una nueva base de datos llamada `Categorias` en la cual se encuentren solamente las variable Cod.establecimiento,N.establecimiento, Palabra.clave, Actividad.economica, Zona en formato tidy

```
Categorias <- baserepilo %>%
                select(Palabra.clave, Actividad.economica,
                       Letrero, Letrero.pintado, Entrada,Zona) %>%
                gather(Variables, Valor, Actividad.economica,
                       Letrero, Letrero.pintado,Entrada)
```

5.2 Listas despegables: Se crean dos listas despegables, una llamada `categoria` la cual posee las variables anteriormente mencionadas y otra lista llamada `palabra_clave` que contiene el tipo de establecimiento comercial guardados en la variable `palabra_clave1`.

```
selectInput("categoria","Categoria:", choices = unique(Categorias$Variables))
selectInput("palabraclave","Tipo de Establecimiento",
            unique(palabra_clave1$Palabra.clave))
```

5.3 Botones: Se crean dos botones que cambian la base `Categorias` por tipo de Zona ya sea Fisdeco o CPS.

```
radioButtons("metodo1", "Metodo:", c("All" = "all","Zona" = "zona"))
```

5.4 Gráfica 1: Se crea la gráfica de barras teniendo en cuenta la lista llamada `categorias` y los botones `all` y `zona`.

```
plotlyOutput("plot", height = 650) # Salida de la gráfica

output$plot <- renderPlotly({

        base2 <- Categorias %>%
                dplyr::filter(Variables == input$categoria) %>%
                count(Palabra.clave,input$categoria,Valor) %>%
                mutate(Palabra.clave= fct_reorder(Palabra.clave, n, sum))
        #Se crea la base2 para que se visualice todos los datos
        base3 <-Categorias %>%
                dplyr::filter(Variables == input$categoria) %>%
                count(Zona, Palabra.clave,input$categoria,Valor) %>%
                mutate(Palabra.clave= fct_reorder(Palabra.clave, n, sum))
        #Se crea la base3 para que se visualice la gráfica por zonas

        plot1 <- function(base){
                ggplot(base, aes(n,Palabra.clave, fill = Valor))+
                geom_col(col= "black")+
                scale_fill_brewer(" ",palette = "Reds", direction = -1)+
                labs(x = "Numero de Establecimientos", y= " ")+
                theme_bw()+
                theme(legend.position = "bottom", legend.box = "vertical",
                panel.background = element_rect(fill = "transparent"),
                plot.background = element_rect(fill = "transparent", color = NA),
                axis.text.x = element_blank(),
                panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                legend.background = element_rect(fill = "transparent"),
                legend.box.background = element_rect(fill = "transparent"),
                legend.title = element_blank())
        } # Se crea una función llamada base que grafique el número de establecimientos
          # por palabra clave filtrado por valor con su respectivo estilo.

 switch(input$metodo1,
        all = plot2 <- plot1(base2), # grafica la función con la base2
        zona = plot2 <- plot1(base3) + facet_grid(.~Zona))
        # grafica la función con la base3 divido por zona

ggplotly(plot2) %>%
        layout(legend = list(x= 0.8,y = 0.01))
        # Se arregla la posición de la leyenda en la parte inferior derecha
         })
```

5.5 Gráfica 2: Se crean tres graficas tipo torta que muestran el porcentaje del tipo de letrero, tipo de entrada y el letrero por `Tipo de establecimiento` sujetos a la lista `palabra_clave`.

```
plotlyOutput("TEntrada",width = "300px",height = "200px") #Salida de la gráfica
output$TEntrada <- renderPlotly({
        tdb_entrada <-baserepilo %>%
                filter(Palabra.clave == input$palabraclave) %>%
```

```
                count(Palabra.clave, Entrada)%>%
                mutate(porcentaje = n/sum(n))
        #Crea la base tdb_entrada que contiene el porcentaje por tipo de establecimiento
        tpentrada <- plot_ly(tdb_entrada, labels = ~Entrada, values =  ~n,
               marker = list( colors = c('rgb(255, 223, 211)','rgb(143, 10, 10 )')),
               type = "pie",showlegend = FALSE) %>%
                layout(title = list(text = "Tipo de Entrada",y= 0.01,x=0.55),
                       plot_bgcolor  = "rgba(0, 0, 0, 0)",
                       paper_bgcolor = "rgba(0, 0, 0, 0)",
                       fig_bgcolor   = "rgba(0, 0, 0, 0)")
        #Se crea el gráfico tpentrada
})
```

```
plotlyOutput("TP_Letrero",width = "300px",height = "200px")
output$TP_Letrero <- renderPlotly({
        tdb_letrero <-baserepilo %>%
                filter(Palabra.clave == input$palabraclave) %>%
                mutate(Letrero = ifelse(Letrero == "si","Posee letrero",
                                        ifelse(Letrero == "no","No posee letrero",0))) %>%
                count(Palabra.clave, Letrero)%>%
                mutate(porcentaje = n/sum(n))
        #Se crea la base tdb_entrada que contiene el porcentaje por tipo de
        #establecimiento
        tpletrero <- plot_ly(tdb_letrero, labels = ~Letrero, values =  ~n,
               marker = list( colors = c('rgb(255, 223, 211)','rgb(143, 10, 10 )'))
               ,type = "pie",showlegend = FALSE) %>%
                layout(title = list(text ="Letrero",y= 0.01,x=0.55),
                       plot_bgcolor  = "rgba(0, 0, 0, 0)",
                       paper_bgcolor = "rgba(0, 0, 0, 0)",
                       fig_bgcolor   = "rgba(0, 0, 0, 0)")
        #Se crea el gráfico tpentrada tpletrero
})
```

```
plotlyOutput("TP_PLetrero",width = "300px",height = "200px")
output$TP_PLetrero <- renderPlotly({
        tdb_pletrero <-baserepilo %>%
                filter(Palabra.clave == input$palabraclave) %>%
                mutate(Letrero.pintado =
                         ifelse(Letrero.pintado == "si",
                                "Posee letrero pintado",
                               ifelse(Letrero.pintado== "no",
                                      "No posee letrero pintado",0))) %>%
                count(Palabra.clave, Letrero.pintado)%>%
                mutate(porcentaje = n/sum(n))
        #Se crea la base tdb_entrada que contiene el porcentaje por tipo de establecimiento
        tpletrero <- plot_ly(tdb_pletrero, labels = ~Letrero.pintado, values =  ~n,
               marker = list( colors = c('rgb(255, 223, 211)','rgb(143, 10, 10 )'))
               ,type = "pie",showlegend = FALSE) %>%
                layout(title = list(text ="Letrero pintado",y= 0.01,x=0.55),
                       plot_bgcolor  = "rgba(0, 0, 0, 0)",
                       paper_bgcolor = "rgba(0, 0, 0, 0)",
                       fig_bgcolor   = "rgba(0, 0, 0, 0)")
        #Se crea el gráfico tpletrero
})
```

6 Tercera Pagina (Datos)

Para finalizar con el dashboard se incluye la `baserepilo` con la opción de descarga

```
datatable(
  baserepilo,
  extensions = 'Buttons', options = list(
    dom = 'Bfrtip',
    buttons =
      list('copy', 'print', list(
        extend = 'collection',
        buttons = c('csv', 'excel', 'pdf'),
        text = 'Download'
      ))

  )
)
```