January 24, 2025

La morfometría fluvial es de gran importancia para realizar estudios hidrológicos en un contexto de cambio climático con amenazas cada vez más frecuentes de eventos extremos, pero aún en la actualidad pocos países realizan de manera continua análisis de esta índole, en especial en nuestro país que, debido a su condición como isla y su ubicación en la trayectoria de ciclones tropicales, debería de estar especialmente interesado en mitigar el riego de grandes inundaciones. A lo largo de este artículo se estará analizando la microcuenca del arroyo caña, perteneciente a la subcuenca del río Macasía ubicada en el extremo suroeste de la República Dominicana, dicha unidad hidrográfica es prácticamente desconocida, ya que no se posee información específica sobre ella en fuentes bibliográficas, por lo que este estudio proporciona algunos datos reales sobre sus aspectos morfométricos, geológicos y cartográficos. Basándose en datos preexistentes y utilizando un MDE se delimitó la cuenca y se analizó la red de drenaje y los perfiles longitudinales e indices de concavidad del curso más largo, de la misma forma se calcularon los parámetros morfométricos de la cuenca, se estudió su posible relación con la litología de la zona y si existían evidencias de un posible fenómeno de reorganizamiento del drenaje.

# Introducción

A lo largo del último siglo se ha reducido la dificultad para realizar análisis espaciales, gracias a los novedosos avances tecnológicos en el desarrollo de los Sistemas de Información Geográfica (SIG) se ha simplificado el arduo trabajo que suponía llevar a cabo este tipo de análisis, y a pesar de todas las herramientas disponibles, la República Dominicana aún está pasos por detrás de muchos países en especial en lo relacionado con los análisis de geomorfología fluvial, situación lamentable, ya que la isla posee innumerables cursos fluviales permitiéndole ocupar un lugar privilegiado en este siglo, ya que, cada día más países sufren por la escasez de agua dulce potable producto del inadecuado manejo de sus cuencas hidrográficas.

La geomorfología fluvial se encarga de estudiar las relaciones entre los procesos físicos del flujo en canales de lecho móvil, la mecánica del transporte de sedimentos forzado por el flujo y las formas de los canales aluviales creadas por el transporte de sedimentos (Elorza, 2008). Esta ciencia posee un amplio campo de estudio, pero en esta investigación nos centraremos en la “morfometría fluvial” la cual se encarga de analizar los parámetros morfométricos de una cuenca hidrográfica, tales como, la red de drenaje, la pendiente, la forma, el orden de la red y demás aspectos físicos. Hasta hace algunos años para analizar los parámetros morfométricos de cualquier sistema fluvial, se requerían una serie de herramientas que hoy en día ya no son enteramente necesarias debido a los números programas que simplifican este arduo trabajo, sin mencionar el tiempo que requería llevar a cabo estos estudios.

Como se menciona con anterioridad se utilizan algunos indicadores para determinar las características de un sistema fluvial, siendo uno de estos el aspecto general del sistema, el cual se entiende como la forma en la que se distribuyen los cursos de agua, esta forma depende principalmente de la gravedad y la pendiente; diversos autores ha establecido métodos tanto cualitativos como cuantitativos para determinar que forma presenta la cuenca, además de que han establecido clasificaciones para denominar las formas similares (ej.: Dendrítica). Cuando nos referimos a la red de drenaje estamos refiriéndonos a la relación entre la longitud total de los cursos fluviales de todos los órdenes y el área de la cuenca, esta variable nos permitirá establecer las características litológicas del área de estudio (Elorza, 2008).

Además, debemos tomar en cuenta, el orden de red de los cursos del sistema, el cual indica el grado de ramificación de la red fluvial; existen distintos métodos para jerarquizar los cursos de una red, pero los dos más conocidos y utilizados son el método de Strahler (1952) y el de Horton (1945), gracias a esta jerarquización se puede entender mejor el comportamiento del sistema de drenaje de la cuenca, además de que se puede obtener la razón de bifurcación descrita por Horton como la relación entre el número de cursos de un orden y número de cursos de orden más alto, esta propiedad es condicionada por la forma que presenta la cuenca, dicho valor debe ser constante de un par de órdenes a otro (Elorza, 2008 Lux Cardona (2016) Ibañez Asensio, Moreno Ramón, & Gisbert Blanquer (2011)).

Gutierrez-Elorza (2008), sostiene que otro elemento utilizado para estudiar las cuencas hidrográficas es el perfil longitudinal de su curso principal, siendo este la línea obtenida a partir de las diferencias de alturas desde el afloramiento del curso hasta la desembocadura en otro cuerpo de agua, este perfil comúnmente es cóncavo, aunque no todos los ríos lo presentan de una manera clara producto de afloramientos de rocas duras, actividad tectónica reciente o debido a cambios súbitos del caudal (Elorza, 2008). A partir del Índice de Concavidad se observa si la cuenca en cuestión presenta realmente un perfil cóncavo, en caso de no serlo es necesario identificar las posibles causas y determinar si existe evidencia de una posible reorganización de la red de drenaje del sistema fluvial.

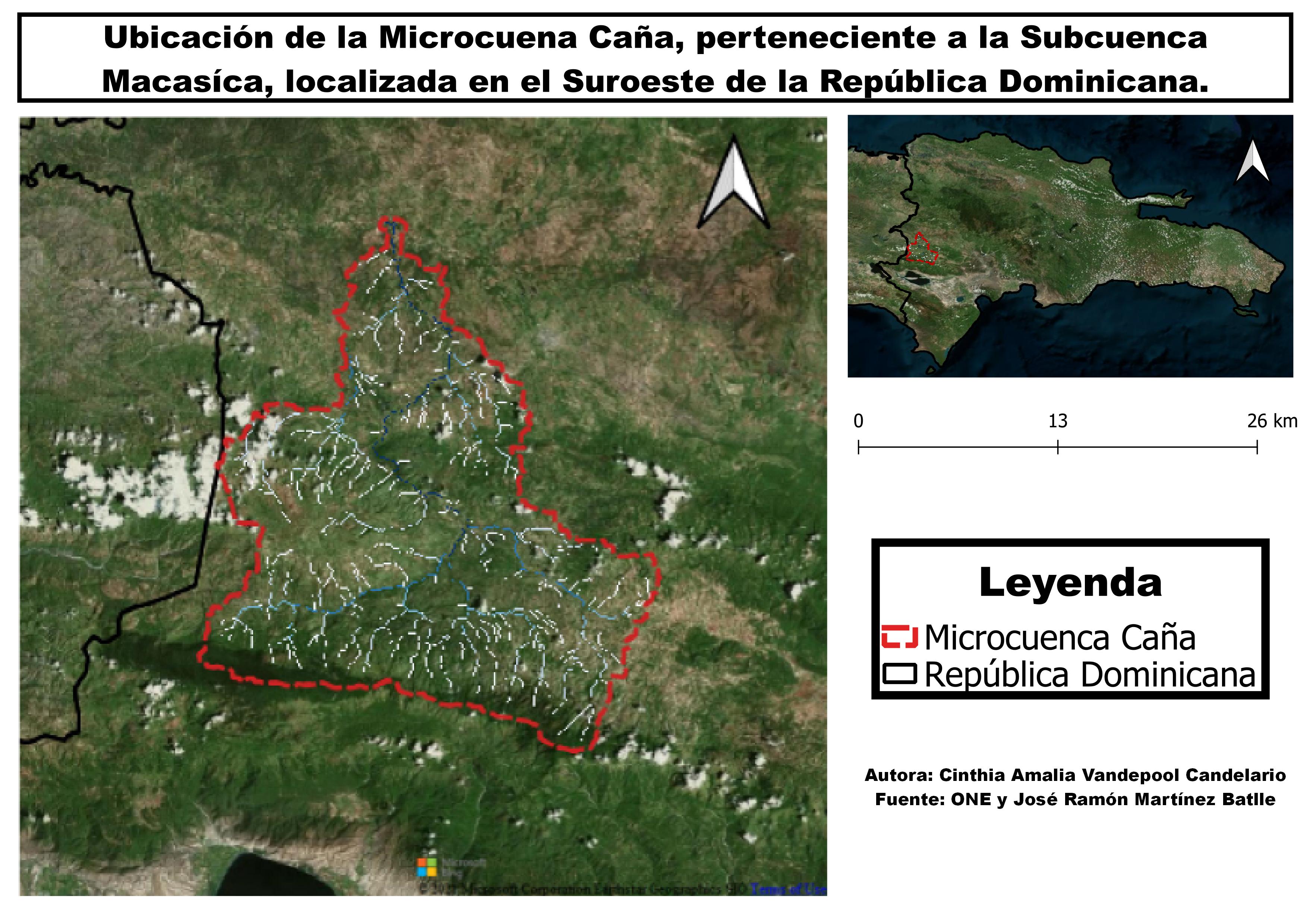
A partir del estudio de los parámetros antes mencionados, se pretende determinar si existe alguna relación entre la forma que presenta la red de drenaje con la litología de la zona, además si existe alguna relación entre las rocas predominantes en la zona (especialmente la caliza) y el fenómeno de reorganizamiento fluvial, de la misma forma se pretende determinar si existe fallas geológica que afectaron directamente a la distribución actual de la red. En mismo orden, se requiere calcular la razón de bifurcación y determinar si esta cumple con lo propuesto por Horton (1945) y de lo contrario analizar su relación con el clima y litología del área de estudio. Con relación a los perfiles longitudinales, se estudiará su relación con la zona por la que discurre el curso y su relación con algún fenómeno geológico en la zona.

Este estudio, por lo tanto, busca aportar esta serie de datos a la comunidad científica dominicana sobre la microcuenca Caña con el objetivo de que puedan ser utilizados en futuros estudios hidrológicos en un contexto de cambio climático con amenazas cada vez más frecuentes de eventos extremos y sus posibles incidencias en las poblaciones asentadas de este singular y escasamente estudiado espacio del suroeste dominicano.

# Metodología

## Área de Estudio

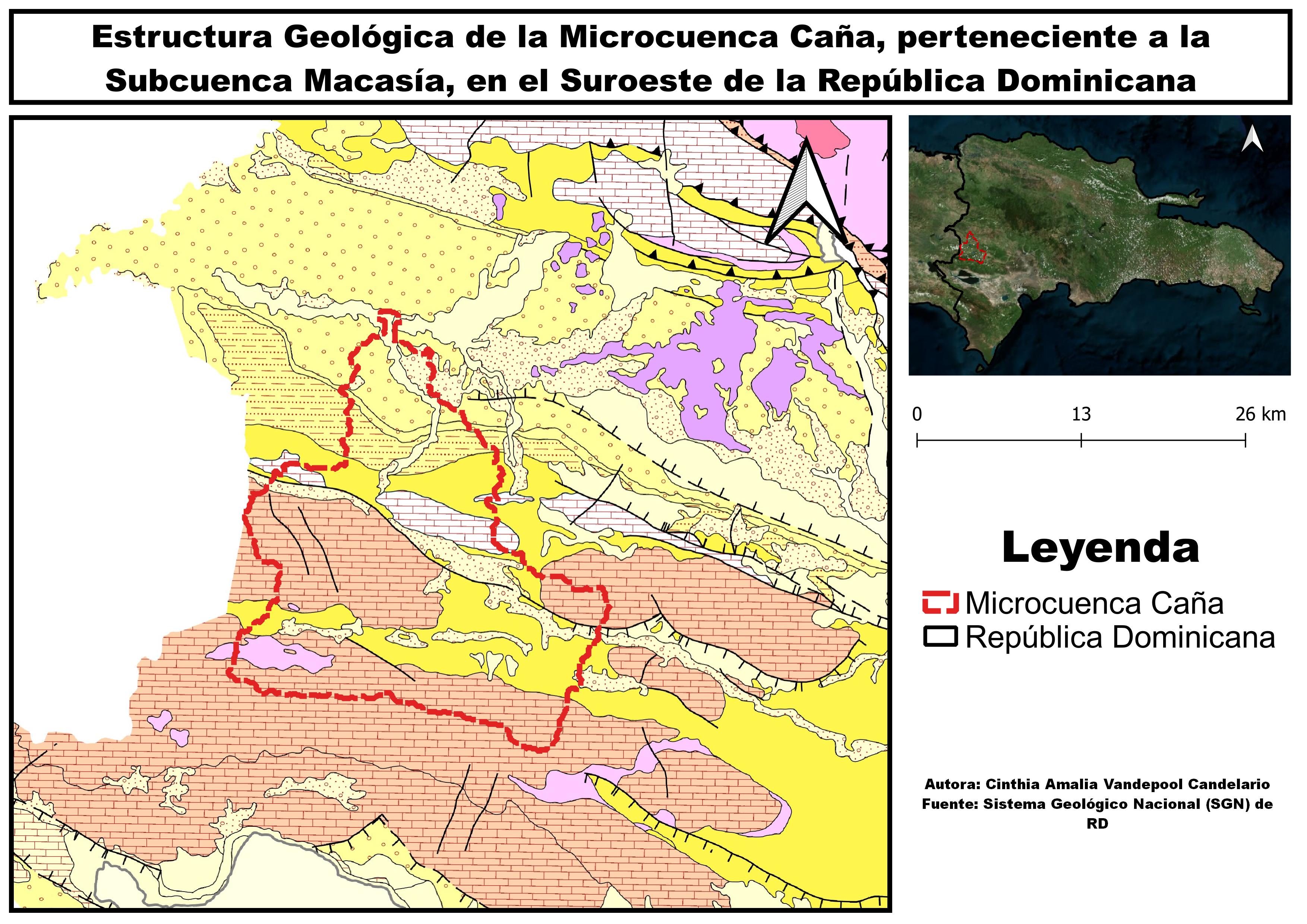
El río Caña nace en la vertiente Norte de la Sierra de Neiba aproximadamente a unos 1,400 metros sobre el nivel del mar. Respecto a su división político-administrativa, la microcuenca del río Caña abarca los municipios de El Cercado y Las Matas de Farfán en la provincia de San Juan y las comunidades de El Llano, Juan Santiago y Hondo Valle de la provincia de Elías Piña. Geográficamente, se localiza entre las coordenadas 18 56’ 25.32" N y 18 37’ 39.64" N latitud norte y 71 27’ 18.45" W y 71 44’ 03.63" W longitud oeste (Ministerio de Medio ambiente y Recursos naturales, 2016) (figura ).



Ubicación Microcuenca del Rio Caña

De acuerdo al mapa Zonas de Vida (OEA, 1967), la mayor superficie de la cuenca lo ocupa el Bosque húmedo subtropical, se caracteriza por presentar topografía que varía desde plana hasta accidentada con un patrón de lluvia que varía de 1000 mm. a 2000 mm. Según la ubicación de las áreas, la biotemperatura media anual es de 23ºC a 24ºC con una evapotranspiración potencial estimada en promedio de 20% menor que la precipitación media total anual. El Bosque muy húmedo Montano Bajo es la segunda en extensión, se caracteriza por la presencia de escarchas temporales, precipitaciones que alcanzar cantidades mayores a los 2,000 mm. totales anuales con una evapotranspiración potencial estimada en promedio de 55% menor que la precipitación media total anual, su topografía generalmente accidentada con elevaciones que van desde los 850 hasta los 2,100 metros y en menor proporción lo ocupa el bosque húmedo montano bajo (Ministerio de Medio ambiente y Recursos naturales, 2016).

La mayor parte de la cuenca se localiza sobre la vertiente Norte del sistema geomorfológico de la Sierra de Neiba y, en menor proporción, sobre el Valle de San juan, siendo la geología conformada, en mayor proporción, por caliza tipo Neiba, marga con calcarenita tipo sombrerito, marga con intercalaciones de bancos de caliza arenosa, arenisca, marga arenosa, conglomerados, conglomerados poligénico, molasa marina y continental y arena; y en menor proporción está conformada por caliza en bancos de espesores variables con nódulos e intercalaciones de pedernal de color blanco-crema, depósitos fluviales, depósitos cuaternarios indiferenciados, basaltos, tobas, aglomerados y rocas volcánicas submarinas (Ministerio de Medio ambiente y Recursos naturales, 2016) (Ver figura )



Estructura Geológica de la Microcuenca Caña

## Materiales, fuentes de datos y métodos

Para la elaboración de esta investigación se emplearon métodos de análisis morfométrico a partir de un modelo digital de elevación (MDE) de la cuenca de interés, el cual es un modelo simbólico, de estructura numérica y digital que pretende representar la distribución espacial de la elevación del terreno, siendo la altura una variable escalar que se distribuye en un espacio bidimensional (Burgos & Salcedo, 2014); inicialmente se cargó una serie de paquetes de R, con el objetivo de adecuar el entorno para la ejecución de los códigos necesarios, entre los que se destacan rgrass7, mapview, leaflet, tidyverse, entre otros [@,]. El paquete rgrass7 se situó encima del software GRASS GIS (GRASS Development Team, 1998–2021), siendo este último el que se ejecutó en el *backend* los algoritmos requeridos para producir los resultados morfométricos, a continuación se detallaran los procedimientos implementados.

En primer lugar, se importó a R, como SpatialGridDataFrame, un MDE (Burgos & Salcedo, 2014) alojado en la base de datos de GRASS GIS (GRASS Development Team, 1998–2021), se estableció su ruta y convirtiéndolo a su vez en un objeto raster por medio del paquete raster de R; partiendo del complemento *r.watershed* (el cual genera un conjunto de mapas que indican: *la acumulación de flujo, la dirección del drenaje, la ubicación de los arroyos y las cuencas hidrográficas* (GRASS Development Team, 2003g)) y del MDE se generaron diversas capas calculando así los parámetros hidrográficos de la cuenca del río caña y sus redes de drenaje, además, seguido a esto se importó un conjunto de capas ráster de GRASS GIS a R, como el mapa de red de drenaje y el mapa de cuencas visualizándolas por medio de *leaflet*.

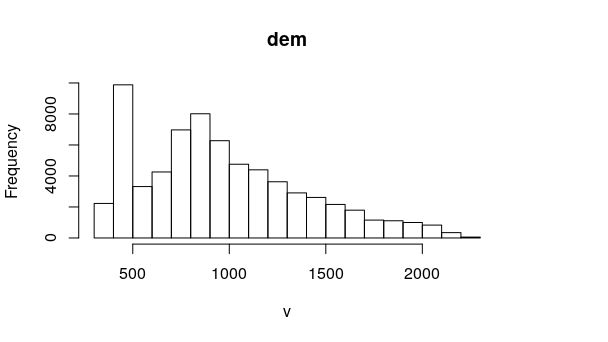
Utilizando el complemento de GRASS GIS *r.water.outlet* (GRASS Development Team, 2003f) y apoyándose en los paquetes *mapview* (Tim Appelhans and others, 2020) y *leaflet* se extrajo la cuenca de drenaje a partir de un mapa de dirección de flujos con un umbral de acumulación de *80 celdas* y las coordenadas de la desembocadura de la cuenca cana (-71.62524,18.94026).

Posteriormente se estableció una máscara usando el límite de la cuenca Caña para luego realizar la extracción partir del MDE de la red de drenaje utilizando el complemento de GRASS GIS *r.stream.extract* (GRASS Development Team, 2003d) desde R. Tras esto, se utilizó el complemento *r.stream*(GRASS Development Team, 2003e) para generar un mapa de dirección de flujo, *r.stream.order* (GRASS Development Team, 2003b) para un mapa de orden de red según varios métodos, entre ellos el método de Strahler y de Horton, a partir de *r.stream.basins* (GRASS Development Team, 2003c) un mapa de cuencas según órdenes de red y apoyándose del complemento *r.stream.stats*(GRASS Development Team, 2003a) se generó las estadísticas de red resumidas por órdenes, incluyendo la razón de bifurcación.

Por último, obtuvimos el curso más largo de la microcuenca a partir de la función *LfpNetwork* creada por José Ramón Martínez (Batlle, 2021), con la misma función, además, pudimos obtener los cursos más largos de las cuencas tributarias. Seguido a esto se generaron los perfiles longitudinales de la microcuenca, se calcularon sus índices de concavidad y a partir de las herramientas *QGIS* y *Google EARTH* cruzamos la información geológica con los patrones de concavidad/convexidad.

# Resultados

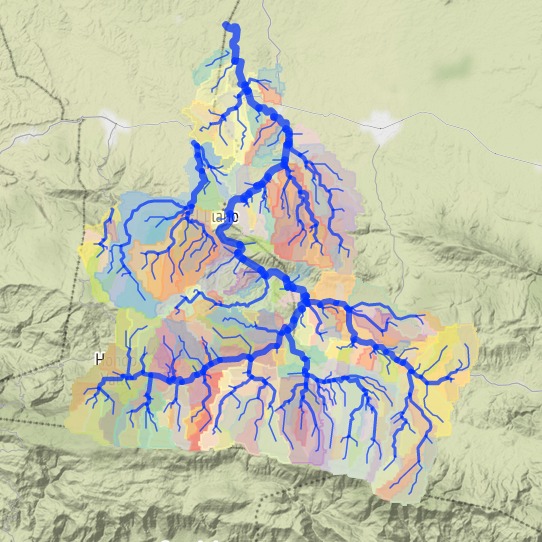
A partir de los códigos ejecutados determinamos que la microcuenca del río caña posee una superficie de 525 km con un perímetro de 139 km, presentando una forma similar a la de un triángulo, presenta mayor extensión en el Sur reduciendo su extensión así el Norte. (figura ). Esta microcuenca posee una elevación máxima de 2,231 metros sobre el nivel del mar, una elevación mínima de 330 metros sobre el nivel del mar y una elevación media de 958 metros. Además, presenta una pendiente media de 10.56 (figura figura ).



Pendiente

La red de drenaje de la microcuenca Caña posee longitud de 445 km distribuidos en 271 cursos de fluviales con una densidad de drenaje de 0.82 km/km, además el orden máximo alcanzado fue de 5.(figura ) (Ver tabla). La mayor parte de la red discurre por caliza, marga y conglomerados; En el tramo alto de la cuenca la red de drenaje es más ancha, concentrándose en esta zona la mayor parte de los cursos mientras en la zona intermedia la red se encoge, disminuyendo también el número de cursos fluviales.

Esta red presenta un patrón de drenaje mixto siendo predominantemente un patrón detrítico en la cuenca media, rectangular en parte alta y baja y paralelo en algunas zonas de la cuenca media. En la zona en donde se presenta un patrón detrítico coincide con calizas, margas y basalto, en la zona en donde predomina el patrón de drenaje rectangular se encuentran rocas de tipo caliza de la formación Neiba y conglomerados mientras en la zona que se encuentra un patrón paralelo predominan las margas y las areniscas.

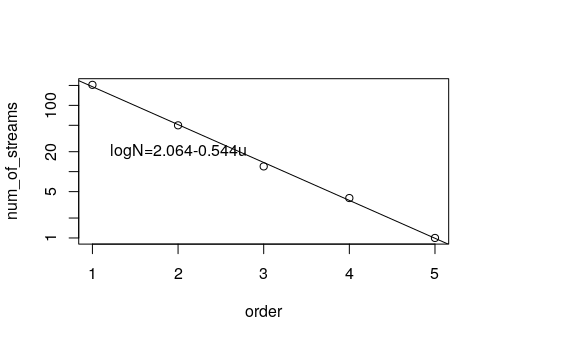


Red de drenaje de la microcuenca Caña

Número de Cursos fluviales en función del Orden

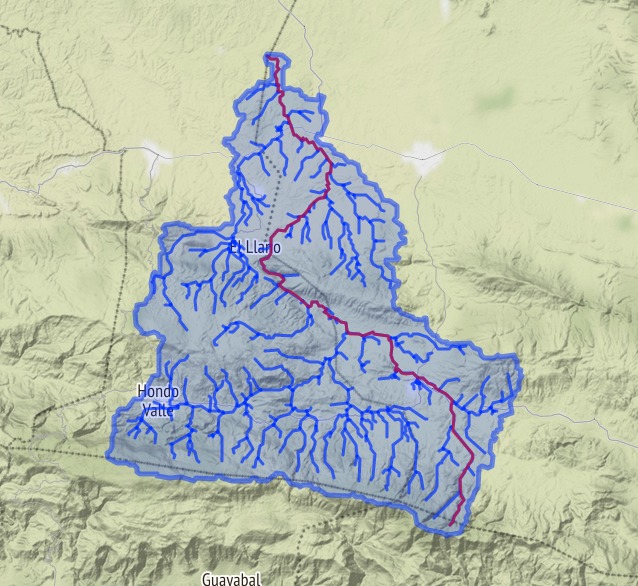
|  |  |
| --- | --- |
| Orden de Red | No. de Cursos fluviales |
| 1 | 204 |
| 2 | 50 |
| 3 | 12 |
| 4 | 4 |
| 5 | 1 |

A continuación, se indicarán la razón de bifurcación de todos los órdenes de red de la microcuenca caña, utilizando dos procedimientos distintos: La razón de bifurcación promedio del par de órdenes 1|2 es 204/50=4.08, para el par de órdenes 2|3 es 50/12=4.16, para el par de órdenes 3|4 es 12/4=3 y para el par de órdenes 4|5 es 4/1=4; el valor promedio es Rb= 3.811667. Mientras que la razón de bifurcación por medio de coeficientes de regresión es Rb= 3.7292 (figura). Además, pudimos observar que las ramificaciones de orden 1 y 2 son las que presentan un grado de bifurcación más alto, las cuales discurren, en mayor proporción, sobre caliza de tipo Neiba y Marga.

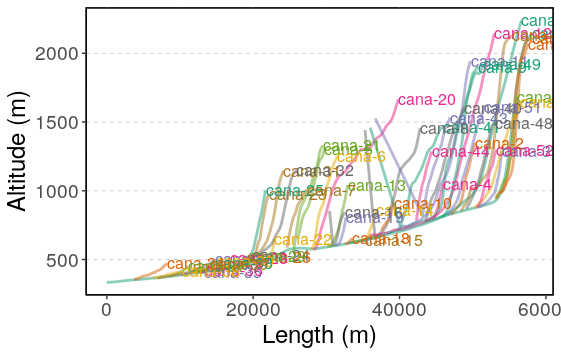


Recta de regresión para el modelo “número de cursos fluviales es función del orden de red” y ecuación correspondiente

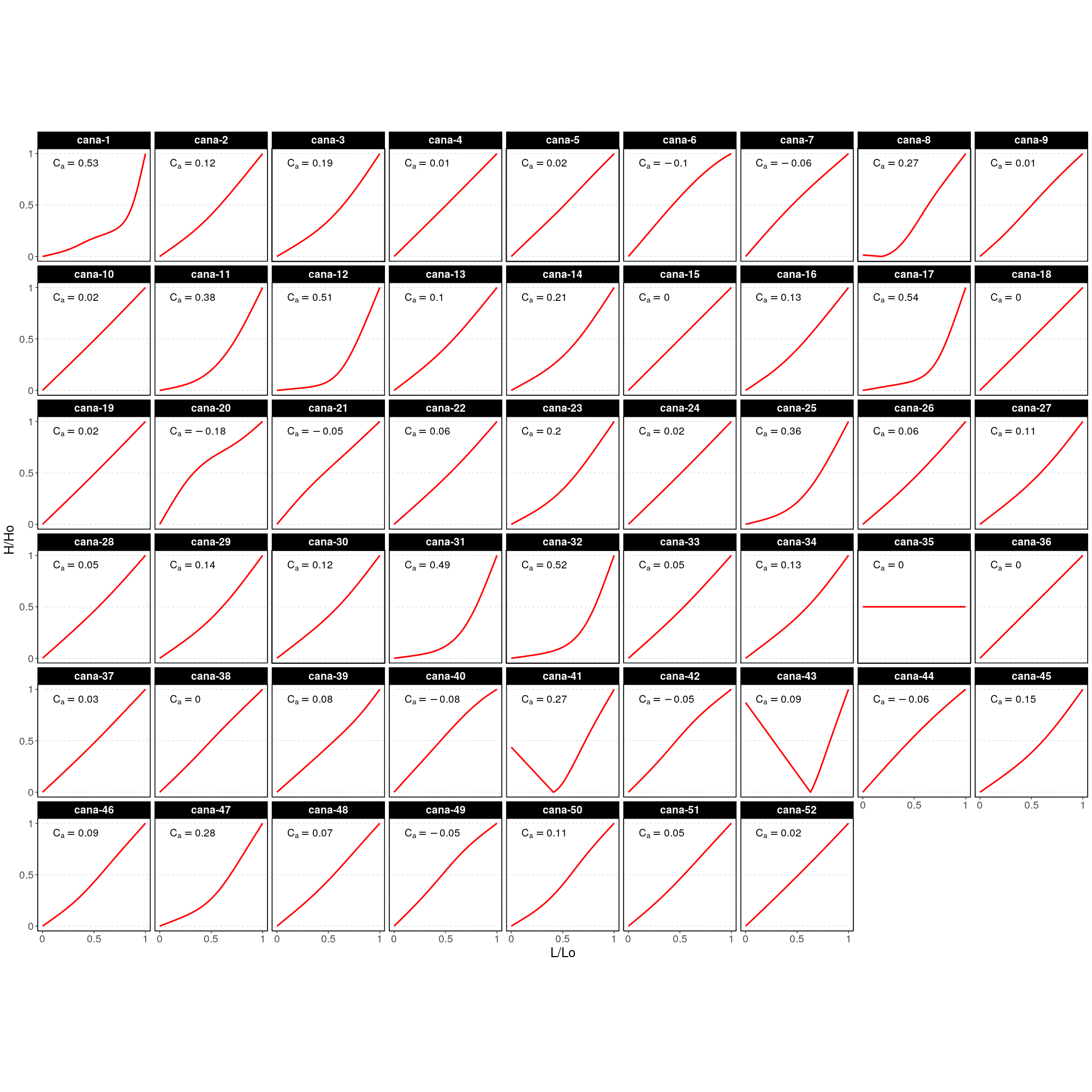
EL curso más largo posee una longitud de 58.1 Kilómetros y discurre por el lado occidental de la cuenca, sobre una falla que atravieza la cuenca en su zona intermedia (figura ). En función de los resultados obtenidos, se observó que la mayor parte de los cursos fluviales de la microcuenca presentan un perfil cóncavo y en pocas ocasiones presentan un perfil convexo, provocado esto por estar en zona de cabecera o por discurrir sobre depósitos aluviales; además, algunos flujos presentan un perfil rectilíneo que también puede deberse al tipo de material por el que discurre.



Curso fluvial más largo de la Microcuenca Caña



Perfil de Concavidad



Indice de Concavidad

A continuacion se resumen algunos parametros morfométricos no mencionados anteriormente: (Ver tabla)

Parametros Morfometricos de la Microcuenca Caña

|  |  |
| --- | --- |
| Parámetros | Valores |
| Orientación Centroide de la cuenca | 226125.00 |
| Norte Centroide de la cuenca | 2075805.00 |
| Rectángulo que contiene la cuenca N-O | (‘211500’, ‘2096370’) |
| Rectángulo que contiene la cuenca S-E | (‘241650’, ‘2061540’) |
| Longitud del vector director [km] | 20.63435024419233 |
| Orientación predominante [grado desde el norte, en sentido contrario a las agujas del reloj] | 1.4441150421051312 |
| Coeficiente de compacidad | 5.390348659271303 |
| Relación de circularidad | 0.33967691382621185 |
| Diámetro topológico | 53.0 |
| Relación de alargamiento | 0.472103656324998 |
| Factor de forma | 9.58527016453617 |
| Tiempo de concentración (Giandotti, 1934) [hr] | 4.981885705233938 |
| Longitud del canal principal [km] | 54.757012947 |
| Pendiente media del canal principal [porcentaje] | 3.57379418847005 |
| Longitud media de la ladera [m] | 152.0461 |
| Magnitud | 147.0 |
| Frecuencia de flujo de primer orden | 0.2800742797000376 |
| Relación de longitud (Horton) | 2.1685 |
| Ratio de superficie (Horton) | 3.9485 |
| Relación de inclinación (Horton) | 1.6639 |

# Discusión

Gracias a las investigaciones y observaciones realizadas se obtuvieron las características morfométricas de la microcuenca Caña. Concretamente se determinó que esta microcuenca presenta un patrón de drenaje detrítico, rectangular y paralelo que según (Elorza, 2008) se debe a la interacción entre el flujo y los materiales erosionables y también se observó que la densidad de drenaje de la microcuenca es baja (0.82 km/km) lo que se interpreta en que los suelos de la cuenca son poco erosionables y presentan escasa capa vegetal (Manuel Córdova, 2016).

También, al superponer la red de drenaje sobre el mapa geológico nacional, se observó que existen la posibilidad de un proceso de reorganizamiento del drenaje, ya que la mayor parte de este curso discurre sobre rocas calizas y margas, de la misma forma se observó que la cuenca es atravesada en su parte intermedia por una falla, que puedo provocar el encogimiento de la red en la parte media-baja. Además, por medio del estudio de la red de drenaje, creemos que el curso principal pudo haber atravesado por un proceso de migración lateral, proceso que pudo ser producto del movimiento de tectónico.

Del mismo modo, se analizó el índice de concavidad de la microcuenca observando que en su mayoría los cursos fluviales presentan un perfil cóncavo, otra parte presentan un perfil convexo y una minoría un perfil rectilíneo; el resultado de los índices cóncavos y convexos se pueden deber a la zona por la que discurre esos cursos ya sea en zona de cabecera o sobre depósitos aluviales, con relación al perfil rectilíneo pocos autores hacen referencia de este por lo que será necesario llevar a cabo nuevas investigaciones.

Con relación a la razón de bifurcación, según Horton (1945) esta razón debe ser constante entre un par de órdenes y el otro, pero en la microcuenca de estudio esto no sucede, aunque estas variaciones fueron mínimas se requerirán estudios más profundos para determinar si esto es producto de las variaciones climáticas o litológicas de esta zona, observando, además, que el cálculo de la razón de bifurcación promedio y la razón de bifurcación por medio de coeficientes de regresión obtuvimos valores distintos, pero con una diferencia pequeña (0.08).

Como consecuencia de los hallazgos de esta investigación se calcularon algunos de los parámetros morfométricos de la microcuenca Caña, sin embargo, este trabajo solo representa la base para formular nuevas hipótesis que requerirán nuevas investigaciones, no solo de esta microcuenca sino también en todos los sistemas fluviales de la isla, ya que este tipo de estudios son escasos o inexistentes en nuestro país.

# Agradecimientos

# Información de soporte

# *Script* reproducible

#parte reutizable del script ----  
#cargar paquetes  
library(rgrass7)  
library(sp)  
library(sf)  
library(raster)  
gisdbase <- 'grass-data-test' #Base de datos de GRASS GIS  
wd <- getwd() #Directorio de trabajo  
wd  
loc <- initGRASS(gisBase = "/usr/lib/grass78/",  
 home = wd,  
 gisDbase = paste(wd, gisdbase, sep = '/'),  
 location = 'cana',  
 mapset = "PERMANENT",  
 override = TRUE)  
  
#unlink\_.gislock()  
  
#Fin de la parte reutilizable  
  
#video 4 ----  
#Definir proyección de la región de GRASS GIS, importar fuente y utilizarla para definir extensión y resolución. Cómo ver la ayuda de las funciones  
#Muestra la definición de la región  
gmeta()  
#Definir ruta del DEM  
dem <- 'datos-fuente/srtm\_dem\_cuenca\_cana.tif'  
  
#Definir la proyección de la región basada en DEM  
execGRASS(  
 cmd = 'g.proj',  
 flags = c('t','c'),  
 georef=dem)  
#Muestra la definición de la región modificada  
gmeta()  
  
#Importar mapa raster  
#r.in.gdal importa la fuente a GRASS  
execGRASS(  
 cmd = 'r.in.gdal',  
 flags=c('overwrite','quiet'),  
 parameters=list(  
 input=dem,  
 output='dem'  
 )  
)  
#Actualizar la extensión de la región al DEM, sólo por precaución  
execGRASS(  
 cmd = 'g.region',  
 parameters=list(  
 raster = 'dem',  
 align = 'dem'  
 )  
)  
#Importar un mapa vectorial también  
demext <- 'datos-fuente/cuenca\_cana.geojson'  
execGRASS(  
 cmd = 'v.in.ogr',  
 flags=c('overwrite','quiet'),  
 parameters=list(  
 input=demext,  
 output='dem\_extent'  
 )  
)  
#Imprimir lista de mapas ráster y vectoriales dentro en la región/localización activa  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
#Ver los addons disponibles en el repositorio oficial de GRASS GIS, incluyendo descripción  
execGRASS(  
 cmd = 'g.extension',  
 flags = 'c'  
)  
  
#Video 5 Explorar datos espaciales básicos entre GRASS y R ----  
#Cargar en R el DEM (mapa ráster)  
use\_sp()  
dem\_sp <- readRAST('dem')  
op <- par()  
plot(dem\_sp)  
  
#Cargar a R el mapa vectorial de una cuenca que se encuentra alojado fuera de GRASS, hacer el plot y representar la cuenca del rio cana superpuesta  
rutacana <- 'datos-fuente/cuenca\_cana.geojson'  
cana <- st\_read(rutacana)  
plot(dem\_sp)  
plot(cana, add=T, col='transparent', border='black', lwd=5);par(op[c('mfrow','mar')])  
  
#Analizar el DEM dentro de la cuenca del rio cana  
dem\_r0 <- raster(dem\_sp)  
dem\_r1 <- crop(dem\_r0, cana)  
dem\_cana <- mask(dem\_r1, cana)  
plot(dem\_cana)  
  
summary(dem\_cana)  
hist(dem\_cana)  
  
#Obtener variables de terreno básicas con el paquete raster dentro de R  
pend\_cana <- terrain(x = dem\_cana, opt = 'slope', unit = 'degrees')  
plot(pend\_cana)

#video 6 Calcular parámetros hidrográficos con r.watershed. Visualizar con leaflet ----  
# Calcular parámetros hidrográficos de interés usando `r.watershed`  
execGRASS(  
 "r.watershed",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 elevation = "dem",  
 accumulation = "accum-de-rwshed",  
 stream = "stream-de-rwshed",  
 drainage = "drainage-dir-de-rwshed",  
 basin = 'basins',  
 half\_basin = 'half-basins',  
 threshold = 80  
 )  
)  
# Traer capas a R  
#Usar Spatial\*  
library(sp)  
use\_sp()  
#Paquete manejo de los raster  
library(raster)  
#DEM  
dem <- raster(readRAST('dem'))  
#Basins  
basins <- raster(readRAST('basins'))  
#Stream network  
stream <- raster(readRAST('stream-de-rwshed'))  
stream3857 <- projectRaster(stream, crs = CRS("+init=epsg:3857"), method = 'ngb')  
#Generar un vectorial de extensión de capa en EPSG:4326  
e <- extent(stream)  
e <- as(e, 'SpatialPolygons')  
proj4string(e) <- CRS("+init=epsg:32619")  
e <- spTransform(e, CRSobj = CRS("+init=epsg:4326"))  
  
# Visualizar capas con `leaflet`  
library(leaflet)  
library(leafem)  
leaflet() %>%  
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addRasterImage(dem, group='DEM', opacity = 0.5) %>%  
 addRasterImage(  
 ratify(basins),  
 group='basins', opacity = 0.7,  
 colors = sample(rep(RColorBrewer::brewer.pal(12, 'Set3'),1000))) %>%   
 addRasterImage(stream3857, project = F, group='str', opacity = 0.7, method = 'ngb', colors = 'blue') %>%   
 addLayersControl(  
 overlayGroups = c('terrain','DEM','basins','str'),  
 options = layersControlOptions(collapsed=FALSE)) %>%   
 addHomeButton(extent(e), 'Ver todo')

#Video 7 Extraer una cuenca de drenaje con r.water.outlet. Visualizar con mapview y leaflet ----  
# Obtener las coordenadas de la desembocadura de la cuenca de interés  
library(mapview)  
mapview(  
 stream3857, method='ngb', col.regions = 'blue',  
 legend = FALSE, label = FALSE, maxpixels = 910425  
)  
# Convertir las coordenadas lat/lon a EPSG:32619  
my\_trans <- function(coords = NULL) {  
 require(sp)  
 pt <- SpatialPoints(matrix(coords, ncol = 2), CRS("+init=epsg:4326"))  
 foo <- spTransform(pt, CRSobj = CRS("+init=epsg:32619"))  
 bar <- as.vector(coordinates(foo))  
 return(bar)  
}  
cana\_out <- my\_trans(coords = c(-71.62524,18.94026))  
cana\_out  
## Extraer la cuenca de interés  
execGRASS(  
 "r.water.outlet",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 input = 'drainage-dir-de-rwshed',  
 output = 'cana-basin',  
 coordinates = cana\_out  
 )  
)  
## Convertir la cuenca a vectorial en GRASS  
execGRASS(  
 "r.to.vect",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 input = 'cana-basin',  
 output = 'cana\_basin',  
 type = 'area'  
 )  
)  
## Mostrar lista nuevamente  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
## Traer a R la cuenca del rio cana  
cana\_bas <- readVECT('cana\_basin')  
cana\_bas  
plot(cana\_bas)  
cana\_bas4326 <- spTransform(cana\_bas, CRSobj = CRS("+init=epsg:4326"))  
leaflet() %>%   
 addProviderTiles(providers$Stamen.Terrain) %>%  
 addRasterImage(stream, opacity = 0.7, method = 'ngb', colors = 'blue') %>%   
 addPolygons(data = cana\_bas4326) %>%   
 leafem::addHomeButton(extent(cana\_bas4326), 'Ver cuenca')  
  
#Video 8 Extraer una red drenaje con r.stream.extract. Visualizar con leaflet ----  
#Usar la cuenca del rio cana como máscara  
execGRASS(  
 "r.mask",  
 flags = c('verbose','overwrite','quiet'),  
 parameters = list(  
 vector = 'cana\_basin'  
 )  
)  
# Extraer la red de drenaje de la cuenca de interés  
execGRASS(  
 "r.stream.extract",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 elevation = 'dem',  
 threshold = 80,  
 stream\_raster = 'cana-stream-de-rstr',  
 stream\_vector = 'cana\_stream\_de\_rstr'  
 )  
)  
# Mostrar lista nuevamente  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
## Traer a R la red de drenaje del rio cana  
cana\_net <- readVECT('cana\_stream\_de\_rstr', ignore.stderr = T)  
cana\_net  
plot(cana\_net)  
cana\_net4326 <- spTransform(cana\_net, CRSobj = CRS("+init=epsg:4326"))  
cana\_net4326  
cana\_centroid <- coordinates(rgeos::gCentroid(cana\_bas4326))  
cana\_centroid  
cana\_net\_r <- raster(readRAST('cana-stream-de-rstr'))  
cana\_net\_r  
cana\_net\_r3857 <- projectRaster(cana\_net\_r, crs = CRS("+init=epsg:3857"), method = 'ngb')  
cana\_net\_r3857  
leaflet() %>%   
 setView(lng = cana\_centroid[1], lat = cana\_centroid[2], zoom = 11) %>%  
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addRasterImage(cana\_net\_r3857, opacity = 0.7, method = 'ngb', colors = 'grey20', group = 'str\_raster') %>%   
 addPolylines(data = cana\_net4326, weight = 3, opacity = 0.7, group = 'str\_vect') %>%   
 leafem::addHomeButton(extent(cana\_net4326), 'Ver todo') %>%   
 addLayersControl(  
 overlayGroups = c('terrain','str\_vect','str\_raster'),  
 options = layersControlOptions(collapsed=FALSE))

#Video 10 Orden de red, morfometría y análisis hortoniano usando r.stream ----  
# Crear mapa de dirección de flujo a partir de r.stream  
execGRASS(  
 "r.stream.extract",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 elevation = 'dem',  
 threshold = 80,  
 direction = 'drainage-dir-de-rstr'  
 )  
)  
#Crear mapas de órdenes de red  
execGRASS(  
 "r.stream.order",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 stream\_rast = 'cana-stream-de-rstr',  
 direction = 'drainage-dir-de-rstr',  
 elevation = 'dem',  
 accumulation = 'accum-de-rwshed',  
 stream\_vect = 'order\_all',  
 strahler = 'order-strahler',  
 horton = 'order-horton',  
 shreve = 'order-shreve',  
 hack = 'order-hack-gravelius',  
 topo = 'order-topology'  
 )  
)  
# Mostrar lista nuevamente  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
# Visualizar la red con leaflet  
#Simbología única  
order <- readVECT('order\_all')  
  
order4326 <- spTransform(order, CRSobj = CRS("+init=epsg:4326"))  
leaflet() %>%   
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addPolylines(  
 data = order4326, weight = 3, opacity = 0.7, group = 'order',  
 label = ~as.character(strahler),  
 highlightOptions = highlightOptions(color = "white",  
 weight = 5, bringToFront = F, opacity = 1),  
 labelOptions = labelOptions(noHide = T,  
 style = list(  
 "font-size" = "8px",  
 "background" = "rgba(255, 255, 255, 0.5)",  
 "background-clip" = "padding-box",  
 "padding" = "1px"))) %>%   
 leafem::addHomeButton(extent(order4326), 'Ver todo') %>%   
 addLayersControl(  
 overlayGroups = c('terrain','order'),  
 options = layersControlOptions(collapsed=FALSE))  
  
#Simbología aplicando grosor según orden de red  
leaflet() %>%   
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addPolylines(  
 data = order4326, weight = order4326$strahler\*1.5, opacity = 0.7, group = 'order',  
 label = ~as.character(strahler),  
 highlightOptions = highlightOptions(color = "white",  
 weight = 5, bringToFront = F, opacity = 1),  
 labelOptions = labelOptions(noHide = F)) %>%   
 leafem::addHomeButton(extent(order4326), 'Ver todo') %>%   
 addLayersControl(  
 overlayGroups = c('terrain','order'),  
 options = layersControlOptions(collapsed=FALSE))  
  
#Delimitar cuencas según orden de red de Strahler  
#Obtener órdenes de red mínimo y máximo  
#Estadísticas para obtener los valores mínimo y máximo del orden de red de Strahler  
rinfo.ordstra <- execGRASS(  
 'r.info',  
 flags = 'r',  
 parameters = list(  
 map = 'order-strahler'  
 )  
)  
#Órdenes de red mínimo y máximo  
minmaxord <- as.numeric(  
 stringr::str\_extract\_all(  
 attributes(rinfo.ordstra)$resOut,  
 "[0-9]+"  
 )  
)  
minmaxord  
  
### Delimitar cuencas, convertirlas de ráster a vectorial  
sapply(  
 min(minmaxord):max(minmaxord),  
 function(x){  
 execGRASS(  
 "r.stream.basins",  
 flags = c('overwrite','c','quiet'),  
 parameters = list(  
 direction = 'drainage-dir-de-rstr',  
 stream\_rast = 'order-strahler',  
 cats = as.character(x),  
 basins = paste0('r-stream-basins-',x)  
 )  
 )  
 execGRASS(  
 "r.to.vect",  
 flags=c('overwrite','quiet'),  
 parameters = list(  
 input = paste0('r-stream-basins-',x),  
 output = paste0('r\_stream\_basins\_',x),  
 type = 'area'  
 )  
 )  
 }  
)  
  
#Representar las cuencas con leaflet  
sapply(  
 min(minmaxord):max(minmaxord),  
 function(x){  
 assign(  
 paste0('orden', x),  
 spTransform(readVECT(paste0('r\_stream\_basins\_',x)), CRSobj = CRS("+init=epsg:4326")),  
 envir = .GlobalEnv)  
 }  
)  
paleta <- RColorBrewer::brewer.pal(12, 'Set3')  
leaflet() %>%   
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addPolygons(data = orden4, stroke = T, weight = 2,  
 color = ~paleta, fillOpacity = 0.4, group = 'O4') %>%   
 addPolygons(data = orden3, stroke = T, weight = 2,  
 color = ~paleta, fillOpacity = 0.4, group = 'O3') %>%  
 addPolygons(data = orden2, stroke = T, weight = 2,  
 color = ~paleta, fillOpacity = 0.4, group = 'O2') %>%  
 addPolygons(data = orden1, stroke = T, weight = 2,  
 color = ~paleta, fillOpacity = 0.4, group = 'O1') %>%  
 addPolylines(  
 data = order4326, weight = order4326$strahler\*1.5,  
 opacity = 0.7, group = 'str\_order') %>%  
 leafem::addHomeButton(extent(order4326), 'Ver todo') %>%   
 addLayersControl(  
 overlayGroups = c('terrain','O1','O2','O3','O4','str\_order'),  
 options = layersControlOptions(collapsed=FALSE))  
  
#Estadísticas de red resumidas por orden de red.  
execGRASS(  
 "r.stream.stats",  
 flags = c('overwrite','quiet','o'),  
 parameters = list(  
 stream\_rast = 'order-strahler',  
 direction = 'drainage-dir-de-rstr',  
 elevation = 'dem',  
 output = 'cana\_stats.txt'  
 )  
)  
file.show('cana\_stats.txt')  
d <- read.csv("cana\_stats.txt", skip=1, header=TRUE)  
plot(num\_of\_streams~order, data=d, log="y")  
mod <- lm(log10(num\_of\_streams)~order, data=d)  
abline(mod)  
text(2, 20, 'logN=2.064-0.544u')  
rb <- 1/10^mod$coefficients[[2]]  
rb  
  
#Estadísticas de red ampliadas  
execGRASS(  
 "r.stream.stats",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 stream\_rast = 'order-strahler',  
 direction = 'drainage-dir-de-rstr',  
 elevation = 'dem',  
 output = 'cana\_stats\_expanded.txt'  
 )  
)  
file.show('cana\_stats\_expanded.txt')

#Video 11 mapview(order, col.regions = 'blue', legend = FALSE) ----  
mapview(order, col.regions = 'blue', legend = FALSE)  
  
# Obtener cursos más largos (cargar función propia)  
devtools::source\_url('https://raw.githubusercontent.com/geofis/rgrass/master/lfp\_network.R') #Cargada como función "LfpNetwork"  
LfpNetwork(  
 xycoords = my\_trans(c(-71.62524,18.94026)),  
 suffix = 'cana',  
 stream\_vect = 'order\_all',  
 direction = 'drainage-dir-de-rstr'  
)  
# Imprimir lista de mapas ráster y vectoriales  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
# Representar con leaflet  
lfp <- readVECT('LfpNetwork\_lfp\_all\_final\_cana')  
lfp4326 <- spTransform(lfp, CRSobj = CRS("+init=epsg:4326"))  
leaflet() %>%  
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addPolylines(  
 data = lfp4326, weight = 3, opacity = 0.7, group = 'order',  
 label = ~as.character(cat),  
 highlightOptions = highlightOptions(color = "white",  
 weight = 5, bringToFront = F, opacity = 1),  
 labelOptions = labelOptions(noHide = T,  
 style = list(  
 "font-size" = "8px",  
 "background" = "rgba(255, 255, 255, 0.5)",  
 "background-clip" = "padding-box",  
 "padding" = "1px"))) %>%   
 leafem::addHomeButton(extent(lfp4326), 'Ver todo')  
  
# Exportar a KML  
execGRASS(  
 'v.out.ogr',  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 input = 'LfpNetwork\_lfp\_all\_final\_cana',  
 output = 'lfp\_kml.kml',  
 format = 'KML',  
 dsco = 'NameField=cat'  
 )  
)  
  
# Obtención de perfiles longitudinales e índices de concavidad  
source('lfp\_profiles\_concavity.R') #Cargado como función "LfpProfilesConcavity"  
cana\_conv\_prof <- LfpProfilesConcavity(  
 xycoords = my\_trans(c(-71.62524,18.94026)),  
 network = 'LfpNetwork\_lfp\_all\_final\_cana',  
 prefix = 'cana',  
 dem = 'dem',  
 direction = 'drainage-dir-de-rstr',  
 crs = '+init=epsg:32619',  
 smns = 1,  
 nrow = 6)  
## Mostrar resultados  
cana\_conv\_prof$profiles  
cana\_conv\_prof$concavityindex  
cana\_conv\_prof$dimensionlessprofiles  
  
## Tabla dx/dy, tanto en metros como adimensional. Útiles para construir perfiles por cuenta propia  
cana\_conv\_prof$lengthzdata %>% tibble::as.tibble()  
cana\_conv\_prof$lengthzdatadmnls %>% tibble::as.tibble()

#Video 12 Parámetros de cuenca con r.basin ----  
# Convertir a números enteros la extensión y la resolución del DEM  
library(raster)  
rutadem <- 'datos-fuente/srtm\_dem\_cuenca\_cana.tif'  
rawextent <- extent(raster(rutadem))  
rawextent  
devtools::source\_url('https://raw.githubusercontent.com/geofis/rgrass/master/integerextent.R')  
devtools::source\_url('https://raw.githubusercontent.com/geofis/rgrass/master/xyvector.R')  
newextent <- intext(e = rawextent, r = 90, type = 'inner')  
newextent  
gdalUtils::gdalwarp(  
 srcfile = 'datos-fuente/srtm\_dem\_cuenca\_cana.tif',  
 dstfile = 'demint.tif',  
 te = xyvector(newextent),  
 tr = c(90,90),  
 r = 'bilinear',  
 overwrite = T  
)  
## Importar a sesión de GRASS  
rutademint <- 'demint.tif'  
execGRASS(  
 "g.proj",  
 flags = c('t','c'),  
 georef=rutademint)  
gmeta()  
execGRASS(  
 "r.in.gdal",  
 flags='overwrite',  
 parameters=list(  
 input=rutademint,  
 output="demint"  
 )  
)  
execGRASS(  
 "g.region",  
 parameters=list(  
 raster = "demint",  
 align = "demint"  
 )  
)  
gmeta()  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
## Generar red de drenaje para obtener coordenada posteriormente  
execGRASS(  
 "r.stream.extract",  
 flags = c('overwrite','quiet'),  
 parameters = list(  
 elevation = 'demint',  
 threshold = 80,  
 stream\_raster = 'stream-de-rstr',  
 stream\_vector = 'stream\_de\_rstr'  
 )  
)  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
## Obtener coordenada  
library(sp)  
use\_sp()  
library(mapview)  
netw <- spTransform(  
 readVECT('stream\_de\_rstr'),  
 CRSobj = CRS("+init=epsg:4326"))  
mapview(netw, col.regions = 'blue', legend = FALSE)  
  
## Transformar coordenada a EPSG:32619 como número entero  
source('my-trans.R')  
outlet <- as.integer(my\_trans(c(-71.62524,18.94026)))  
  
## Ejecutar `r.basin`  
pref <- 'rbasin\_cana'  
execGRASS(  
 "r.basin",  
 flags = 'overwrite',  
 parameters = list(  
 map = 'demint',  
 prefix = pref,  
 coordinates = outlet,  
 threshold = 80,  
 dir = 'salidas-rbasin/cana'  
 )  
)  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
#> Si `r.basin` arrojara error (sólo en el caso de error, no en caso de advertencia), ejecutar este bloque para borrar las salidas anteriores y reejecutar el `r.basin`:  
#execGRASS(  
 # "g.remove",  
 #flags = 'f',  
 #parameters = list(  
 # type = c('raster','vector'),  
 #pattern = paste0(pref, '\*')  
 #)  
#)  
## Cargar los vectoriales transformados a EPSG:4326 para visualizar en leaflet  
rbnetw <- spTransform(  
 readVECT('rbasin\_cana\_demint\_network'),  
 CRSobj = CRS("+init=epsg:4326"))  
rbnetw  
rbmain <- spTransform(  
 readVECT('rbasin\_cana\_demint\_mainchannel'),  
 CRSobj = CRS("+init=epsg:4326"))  
rbmain  
rbbasin <- spTransform(  
 readVECT('rbasin\_cana\_demint\_basin'),  
 CRSobj = CRS("+init=epsg:4326"))  
rbbasin  
  
library(leaflet)  
leaflet() %>%  
 addProviderTiles(providers$Stamen.Terrain, group = 'terrain') %>%  
 addPolylines(data = rbnetw, weight = 3, opacity = 0.7) %>%   
 addPolylines(data = rbmain, weight = 3, opacity = 0.7, color = 'red') %>%   
 addPolygons(data = rbbasin) %>%   
 leafem::addHomeButton(extent(rbbasin), 'Ver cuenca')  
  
## Explorar los parámetros de cuenca  
library(readr)  
rbcanapar1 <- read\_csv("salidas-rbasin/cana/rbasin\_cana\_demint\_parametersT.csv")  
rbcanapar1 %>% tibble::as\_tibble()  
rbcanapar2 <- read\_csv(  
 "salidas-rbasin/cana/rbasin\_cana\_demint\_parameters.csv",  
 skip=2, col\_names = c('Parameter', 'Value'))  
rbcanapar2 %>% print(n=Inf)

#Video 13 Curva e integral hipsométrica ----  
# Imprimir lista de mapas ráster y vectoriales dentro en la región/localización activa  
#\* Nótese que los paquetes requeridos en esta sessión (`rgrass7`, `raster`, `leaflet`, `leafem`), fueron en el bloque anterior al ejecutarse el código contenido en el archivo `orden-de-red.Rmd`. Igualmente, dicho bloque de código creó todos los objetos necesarios para realizar este tutorial.  
execGRASS(  
 'g.list',  
 flags = 't',  
 parameters = list(  
 type = c('raster', 'vector')  
 )  
)  
## Representar cuencas  
library(sp)  
use\_sp()  
library(mapview)  
bas2 <- readVECT('r\_stream\_basins\_2')  
bas3 <- readVECT('r\_stream\_basins\_3')  
  
## Curva e integral hipsométrica  
source('integral\_hypsometric\_curve.R') #Cargada como función "HypsoIntCurve"  
HypsoBasinsOrder2 <- HypsoIntCurve(  
 basins = 'r\_stream\_basins\_2',  
 dem = 'dem',  
 labelfield = 'cat',  
 nrow = 2,  
 labelsize = 4  
)  
  
HypsoBasinsOrder2$HypsoInt  
HypsoBasinsOrder2$HypsoCurve  
mapview(bas2, zcol='cat', col.regions = 'blue', legend = FALSE) %>%  
 addStaticLabels(label = bas2$cat)  
  
HypsoBasinsOrder3 <- HypsoIntCurve(  
 basins = 'r\_stream\_basins\_3',  
 dem = 'dem',  
 labelfield = 'cat',  
 nrow = 1,  
 labelsize = 4  
)  
  
HypsoBasinsOrder3$HypsoInt  
HypsoBasinsOrder3$HypsoCurve  
mapview(bas3, zcol='cat', col.regions = 'blue', legend = FALSE) %>%  
 addStaticLabels(label = bas3$cat)

# Referencias

Batlle, J. R. M. (2021). geomorfologia-master/unidad-4-asignacion-1-procesos-fluviales: Let’s map (Version v0.0.0.9000). <https://doi.org/10.5281/zenodo.4425878>

Burgos, V. H., & Salcedo, A. P. (2014). Modelos digitales de elevación: Tendencias, correcciones hidrológicas y nuevas fuentes de información. *Encuentro de Investigadores En Formación En Recursos Hídricos (2, 2014, Ezeiza, Buenos Aires, Argentina). Disponible En: Http://Www. Ina. Gov. Ar/Ifrh-2014/Eje1/1.11. Pdf. Consultado*, *1*(10), 2015.

Elorza, M. G. (2008). Geomorfologia fluvia i. In *Geomorfologia* (pp. 279–283). Pearson Educacion.

GRASS Development Team. (1998–2021). GRASS gis. Retrieved May 24, 2021, from <https://grass.osgeo.org/>

GRASS Development Team. (2003a). Calculates horton’s statistics for strahler and horton ordered networks created with r.stream.order. Retrieved April 12, 2021, from <https://grass.osgeo.org/grass78/manuals/addons/r.stream.stats.html>

GRASS Development Team. (2003b). Calculates strahler’s and more streams hierarchy. Retrieved April 12, 2021, from <https://grass.osgeo.org/grass78/manuals/addons/r.stream.order.html>

GRASS Development Team. (2003c). Delineates basins according stream network. Retrieved April 12, 2021, from <https://grass.osgeo.org/grass78/manuals/addons/r.stream.basins.html>

GRASS Development Team. (2003d). Performs stream network extraction. Retrieved April 12, 2021, from <https://grass.osgeo.org/grass78/manuals/r.stream.extract.html>

GRASS Development Team. (2003e). R.stream.\* modules. Retrieved April 12, 2021, from [https://grasswiki.osgeo.org/wiki/R.stream.\*\_modules](https://grasswiki.osgeo.org/wiki/R.stream.*_modules)

GRASS Development Team. (2003f). R.water.outlet - creates watershed basins from a drainage direction map. Retrieved April 2, 2021, from <https://grass.osgeo.org/grass78/manuals/r.water.outlet.html>

GRASS Development Team. (2003g). R.watershed - calculates hydrological parameters and rusle factors. Retrieved April 2, 2021, from <https://grass.osgeo.org/grass76/manuals/r.watershed.html>

Ibañez Asensio, S., Moreno Ramón, H., & Gisbert Blanquer, J. M. (2011). *Morfología de las cuencas hidrológicas*.

Lux Cardona, B. (2016). *Conceptos básicos de morfometría de cuencas hidrográficas*.

Manuel Córdova. (2016). Parámetros geomorfológicos de cuencas hidrográficas. Retrieved May 11, 2021, from <https://www.prontubeam.com/articulos/articulos.php?Id_articulo=26>

Ministerio de Medio ambiente y Recursos naturales. (2016). Macasía. Retrieved April 28, 2021, from <https://ambiente.gob.do/cuencas-hidrograficas/macasia/>

Tim Appelhans and others. (2020). Mapview: Interactive viewing of spatial data in r. Retrieved April 12, 2021, from <https://cran.r-project.org/web/packages/mapview/index.html>