



# **R básico para SIG**

¿¿Sin usar R Studio!?!?



# Sobre mí

- Soy Tabasqueña, originaria de Jalpa de Mendez
- Licenciado en Gestión Ambiental
- Analista de datos geoespaciales y SIG
- Primera experiencia con R en noviembre 2020


¿Qué hago actualmente?

- Estudio en un Diplomado "Restauración de ecosistemas y servicios ambientales"
- Abrí un capítulo de R-Ladies en Villahermosa



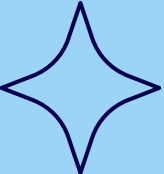




# ¿Qué son los Sistemas de Información Geográfica?

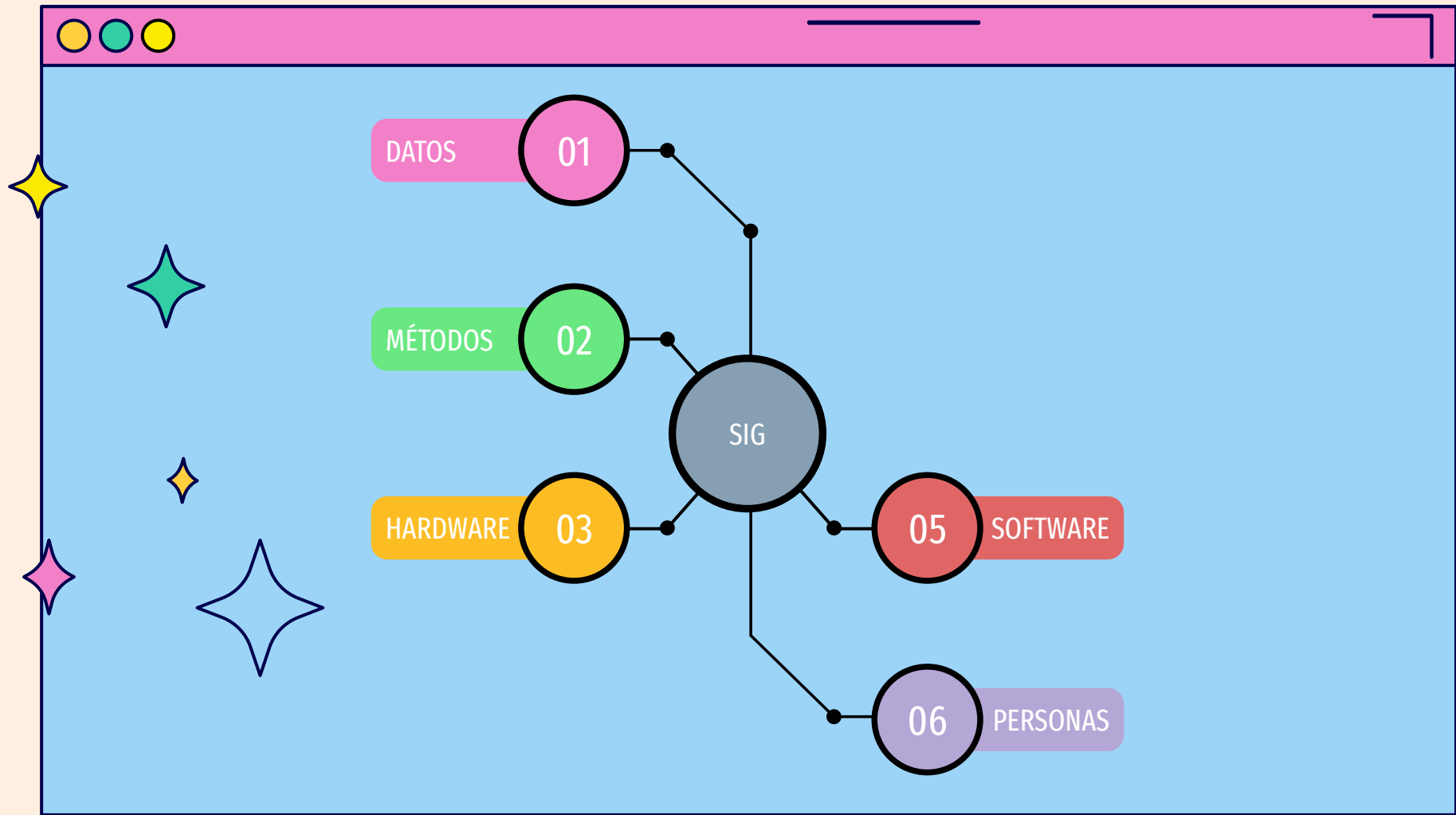


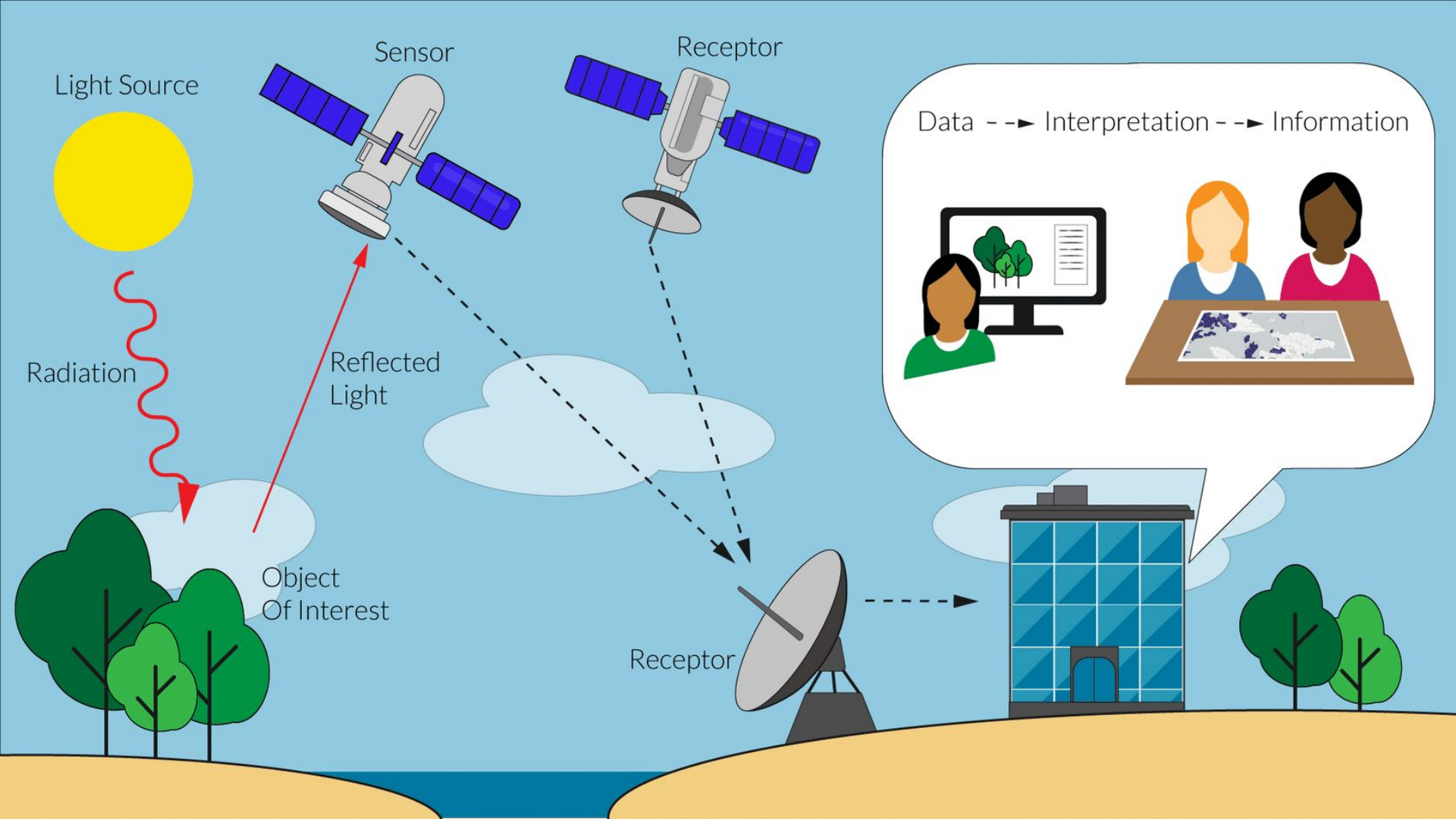
Un sistema de información geográfica (SIG) es un marco para recopilar, gestionar y analizar datos.

Adaptado en la ciencia de la geografía, junto con otras ciencias. Analiza la ubicación espacial y organiza capas de información en visualizaciones utilizando mapas y escenas en 3D.



Con esta capacidad única, revela conocimientos más profundos sobre los datos, como patrones, relaciones y situaciones, lo que ayuda a los usuarios a tomar decisiones más inteligentes.





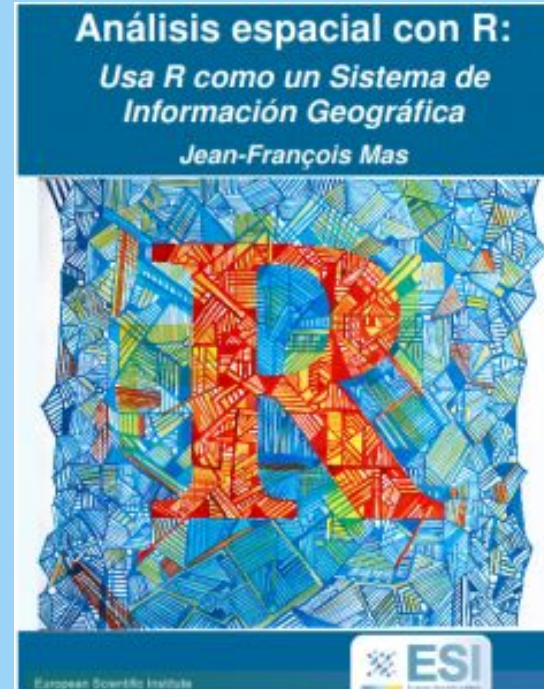
# R aplicado a SIG

El lenguaje de programación R es una potente herramienta para realizar análisis estadístico que también puede solucionar complejos análisis de datos espaciales.

La relación entre R y los Sistemas de Información Geográfica (SIG) tienen muchos antecedentes de trabajos.

La vinculación de los GIS y la Cartografía, la Geoestadística y Ciencia de Datos para información geográfica (o Spatial Data Science) está más presente que nunca, contando cada vez mayor número de especialistas y aplicación a todo tipo de proyectos.

Algunas características de R aplicado a SIG son:





**01**

**Clases para datos  
espaciales**

**02**

**Lectura y escritura de  
datos espaciales**

**03**

**Análisis de patrones de  
puntos**

**04**

**Geoestadística**

**05**

**Regresión espacial**

**06**

**Análisis ecológico**







**07**

**Algoritmos de  
procesamiento ráster**

**10**

**Web Mapping**

**08**

**Teledetección y LIDAR**



**09**

**Detección remota**



# Cualidades iguales

## R en Google Colab

- ✦ Proporciona acceso directo en el navegador a Jupyter Notebook
- ✦ GPU gratis
- ✦ Permite almacenar cuadernos en Google Drive y compartirlos

## RStudio Cloud



- ✦ Tiene la versión de R más reciente
- ✦ Pueden convivir diferentes versiones de R
- ✦ Facilita compartir ambientes de trabajo

[https://nekrum.github.io/NeurocienciasIntroData/rstudio\\_cloud#8](https://nekrum.github.io/NeurocienciasIntroData/rstudio_cloud#8)



# ¿Qué vamos a ver?



- ◆ rgdal, raster y rgeos
- ◆ Modelación numérica para resolver problemas comunes
- ◆ Lectura y escritura de datos
- ◆ Ajuste de modelos
- ◆ Estimación y visualización de datos



# ¿Para qué se usan estas librerías?

rgdal: Lectura y escritura de datos espaciales. Contiene funciones para escribir archivos ráster y vectoriales en formatos compatibles.

rgeos: Interface to Geometry Engine – Open Source ('GEOS') El paquete rgeos proporciona una interfaz para funciones de topología para objetos espaciales de sp utilizando la librería GEOS.

raster: Nos va a permitir la lectura, escritura, manipulación, análisis y modelado de datos espaciales de tipo ráster (gridded spatial data).




# GLOSARIO

**Shapefile:** El Shapefile es un formato de representación vectorial desarrollado por ESRI (Environmental Systems Research Institute). Consta de un número variable de archivos, en los que se almacena digitalmente la localización de los elementos geográficos (archivo shape \*.shp) junto con sus atributos o características (tabla dBase \*.dbf).

**Vector:** Proporciona una manera de representar "objetos espaciales" del mundo real dentro de un ambiente SIG. Un objeto espacial es algo que puede ver en el paisaje. Por ejemplo: casas, carreteras, arboles, ríos etc.

**Hillshade:** Los archivos Hillshade responden a modelos digitales de terreno cuya función es la representación de sombras a lo largo de la geografía del territorio.





# El taller

<https://github.com/jdanifalcon/match-rstudio-colab>

Simplemente abrimos un Notebook en Google Colab y cargamos el paquete con el siguiente comando:

```
%load_ext rpy2.ipython
```

Si todo está OK, ¡enhorabuena! Ya puedes usar R dentro de Python. ¿Cómo? Pues sencillo: cada vez que quieras instalar una paquetería de R, deberás «llevarla» a R usando el siguiente código:

```
%%R  
install.packages("rgdal")  
install.packages("raster")  
install.packages("rgeos")  
library(rgdal)  
library(raster)
```

Así, ya podrás operar con R dentro de Python.





# THANKS!



Si tienes alguna pregunta  
[jdofalcon@protonmail.com](mailto:jdofalcon@protonmail.com)  
[beacons.page/jdofalcon](https://beacons.page/jdofalcon)