

# Proyecto de introducción a los algoritmos

Autor: Lucas Hector Fontini Giorgiutti, Comisión 1

Este es el informe para la materia Introducción a los algoritmos

## Vigilancia de las enfermedades por virus del Dengue y Zika

Elegí la Base de datos del Ministerio de Salud Secretaría de Promoción de la Salud, Prevención y Control de Riesgos Subsecretaría de Prevención y Control de Enfermedades Comunicables e Inmunoprevenibles Dirección Nacional de Epidemiología y Análisis de la Situación de Salud Área de Vigilancia. Que en esta pueden ver los casos confirmados de Dengue y Zika correspondientes al Registro del Sistema Nacional de Vigilancia de la Salud en Julio de 2021 . Esta base de datos se actualiza cada 6 meses y esta fue creada en el 28 de diciembre de 2018 tiene el fin de denotar la evolución que tiene el virus a través del tiempo y las distintas épocas del año, y así de esta forma poder tener un control de la situación del país con respecto de esta enfermedad

Elegí esta base de datos por la versatilidad que creo que se puede obtener con los datos que se dan, y por los interesantes resultados que pueden llegar a obtenerse, como por ejemplo cómo afecta a las distintas regiones, y que generación se vio más afectada. Además una vez hechas las funciones me doy cuenta que la mayoría podrían ser utilizadas para otro virus al que se le dé el mismo formato de base de datos, ya que los campos son muy versátiles y amplios

De aquí fue obtenida la base:

<https://datos.gob.ar/dataset/salud-vigilancia-enfermedades-por-virus-dengue-zika>

## La base tiene este formato:

**(String,String,Int,Int,String,String,Int)**

por ejemplo las primeras 11 tuplas son:

-("San Fernando","Chaco",2021,18,"Dengue","Neonato (hasta 28 dias)",1)  
-("Patiño","Formosa",2021,6,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Patiño","Formosa",2021,8,"Dengue","Posneonato (29 hasta 365 dias)",1)

-("Patiño","Formosa",2021,10,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Capital","La Rioja",2021,12,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Iguazú","Misiones",2021,6,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Anta","Salta",2021,11,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Anta","Salta",2021,16,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Grl. José de San Martín","Salta",2021,14,"Dengue","Posneonato (29 hasta 365 dias)",1)  
-("Orán","Salta",2021,17,"Dengue","Posneonato (29 hasta 365 dias)",1)

## Campos en los que se divide la base

**provincia** : Este campo representa la provincia en la que se está midiendo la cantidad de los casos de los virus

**departamento** : Este campo representa la ciudad en la que se está midiendo la cantidad de los casos de los virus

**año** : Aca se ve el año en el que se midió (2021)

**semanas epidemiológicas** : son lapsos que se cuentan de domingo a sábado en la que se miden los padecimientos epidemiológicos a través del tiempo

**evento nombre**: es el virus que se padeció (En este caso es se ve que solamente fue dengue)

**grupo de edad** : Acá se puede ver el grupo de edad que fue afectado

**cantidad de casos**: Son la cantidad de casos que hubo en la region

## Registro de los campos en haskell

```
type Dengue = (String,String,Int,Int,String,String,Int)

dengue :: [Dengue]
```

### Funciones de haskell:

`provincia :: Dengue -> String`

-- Puesta una tupla de tipo "Dengue" te da la provincia en la que fue hecho el censo

-----1-----

ciudad :: Dengue -> String

-- Puesta una tupla de tipo "Dengue" te da la Ciudad en la que fue hecho el censo

-----2-----

ano :: Dengue -> Int

-- Puesta una tupla de tipo "Dengue" te da el año

-----3-----

semanaEp :: Dengue -> Int

-- Puesta una tupla de tipo "Dengue" te da la cantidad semanas epidemiológicas que pasaron en el lugar especificado

-----4-----

virus :: Dengue -> String

-- Puesta una tupla de tipo "Dengue" te da el virus que afectó a las personas

-----5-----

promDeEdad :: Dengue -> String

-- Puesta una tupla de tipo "Dengue" te da el promedio de edad de las personas afectadas

-----6-----

Personas :: Dengue -> Int

-- Puesta una tupla de tipo "Dengue" te da la cantidad de personas fueron afectadas por el virus (del promedio de edad que especifique la tupla)

-----7-----

`personasPorProv :: [Dengue] -> String -> Int`

Puesta Una lista de tipo [Dengue] Y una provincia, te muestra la cantidad de personas afectadas allí

Por Ejemplo en el caso de introducir la lista dengue y la String Cordoba (sin acento porque haskell no los lee) , nos va mostrar que hubo 6 casos

-----8-----

`tuvoContagiosEstaprov :: [Dengue] -> String -> Bool`

Puesta una la lista tipo [Dengue]y una provincia , te dice si esta tuvo casos o no, en el caso de haber tenido va mostrar "True", en el caso contrario mostrará "False"

-----9-----

`queCiudadHayMas :: [Dengue] -> (String,String)`

En esta función al poner una lista de tipo [Dengue] te mostrará la ciudad en la que hubo más casos en forma de lista, con su respectiva provincia

Por Ejemplo en el caso de la lista "dengue" nos mostrará que es [capital,La Rioja]

-----10-----

`tuvoMasde :: Int -> [Dengue] -> [String]`

En esta función al poner un entero y una lista de tipo [Dengue] , te mostrará en una lista todas las provincias que tuvieran una cantidad de contagios mayores a ese entero

Por ejemplo si ponemos 1000 nos mostrará la lista ["La Rioja, "Salta"]

-----11-----

`totalDeCasos :: [Dengue] -> Int`

Esta Función al poner una lista de tipo [Dengue] nos dará la suma de todas las personas que contrajeron dengue de todas las ciudades

En el caso de la base de datos dengue (que es la de los últimos 6 meses) es un total de 3850 casos

-----12-----

`conRespectoal2020 :: [Dengue] -> [Dengue]`

Esta función sólo servirá con la lista "dengue" ya que lo que hace es que puesta esta lista nos devuelve la cantidad de personas de cada ciudad multiplicada por 12 , ya que hubo 12 veces más casos en 2020 y de esta manera nos permite comparar de una manera estimativa ( la cual es muy poco precisa ya que el año no terminó y el verano es siempre un pico con respecto a la cantidad de casos)

-----13-----

`sacar :: [String] -> String -> [String]`

Esta y la próxima función fueron realizadas para poder hacer la función número 10. Lo que hace la función es puesta una lista de palabras y una palabra te la saca de la lista

-----14-----

`queNoSeRepita :: [String] -> [String]`

Esta función lo que hace es sacar todas las palabras repetidas de una lista

afectadosPorEdad :: [Dengue] -> String -> Int

Esta función dada una lista de tipo [Dengue] y un rango de edad en forma de String ( además tiene que ser parte de una tupla) , te dirá cuántos son los afectados de ese grupo

Por ejemplo si pones la lista "dengue"