

Manual de usuario

Pasos a seguir:

1- En la siguiente imagen podemos observar los diferentes archivos de datos que utilizamos en la aplicación, cómo se puede observar se pueden utilizar tanto archivos csv cómo JSON, los cuales están ubicados en una carpeta nombrada data que se encuentra en la carpeta principal del proyecto.



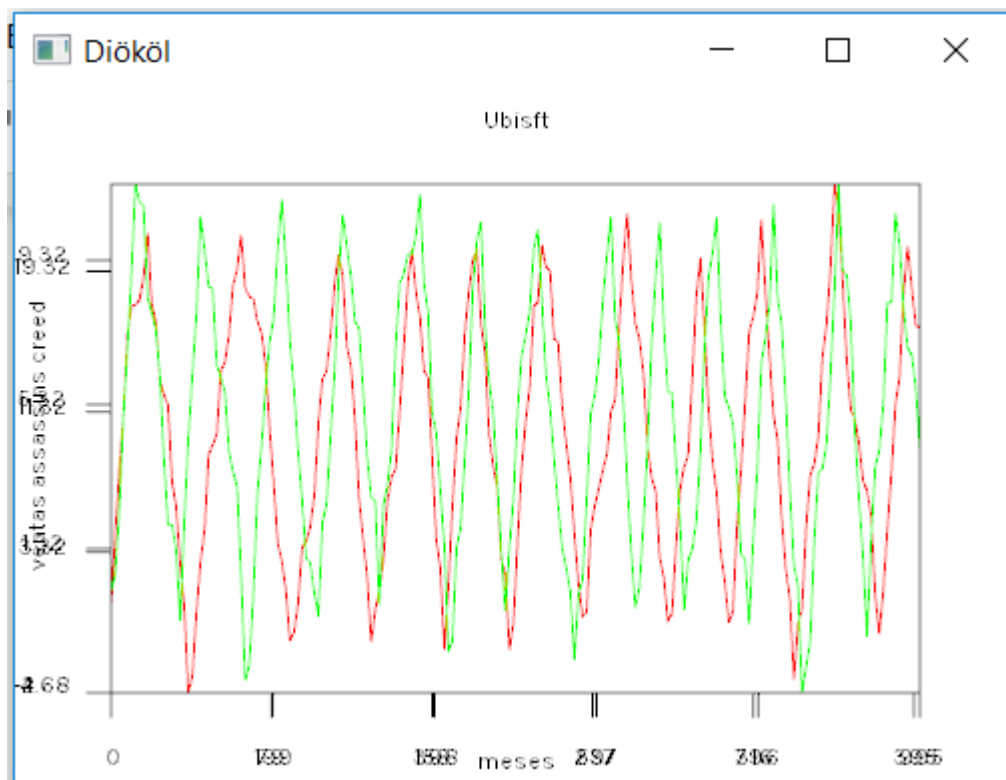
2- En la siguiente imagen, específicamente en la parte marcada podemos observar cómo hacemos la llamada de los archivos, hay que colocar el nombre correcto del archivo y cómo se observa definir que esos archivos están en la carpeta data, además se puede observar cómo se hace la lectura bien sean archivos csv o JSON.

```
28
29 function setup()
30     size(500,350)
31     local f = loadFont("data/Karla.ttf",12)
32     textFont(f)
33     bg = Dk1BaseGraphics:new(width(),height())
34     data = readCSV("data/venta1.csv",true,',')
35     data2 = readCSV("data/venta2.csv",true,',')
36     data3 = readCSV("data/venta3.csv",true,',')
37
38     str = readAll("data/venta1.json")
39     dd = json.decode(str)
40     datax = getColumn(dd,"x")
41     datay = getColumn(dd,"y")
42     str = readAll("data/venta2.json")
43     dd = json.decode(str)
44     datax2 = getColumn(dd,"x")
45     datay2 = getColumn(dd,"y")
46     str = readAll("data/venta3.json")
47     dd = json.decode(str)
48     datax3 = getColumn(dd,"x")
49     datay3 = getColumn(dd,"y")
50 end
51
52 function draw()
53     background(255)
54     --bg:par({mfrow={2,1}})
55     bg:plot(data["x"],data["y"],{type="l",sub="Ubisft",ylab="ventas assassins creed",xlab="meses",col = "#FF0000"})--
56     bg:plot(data2["x"],data2["y"],{type="l",sub="Ubisft",ylab="ventas assassins creed",xlab="meses",col = "#00FF00"})
57 end
58
59 function windowResized(w,h)
60     bg:resize_window(w,h)
61 end
62
```

3- En la siguiente imagen podemos observar la forma en la que se debe ejecutar la aplicación, debemos ubicarnos en la carpeta principal del proyecto y abrir una ventana del cmd desde esa dirección, estado ahí en esa ventana que abrimos debemos escribir lo siguiente Diokol.exe proyecto.lua y seguidamente debemos oprimir enter, esto ejecutará automáticamente el gráfico que se genera a partir de la lectura de los archivos.

```
C:\Windows\System32\cmd.exe - diokol.exe proyecto.lua
Microsoft Windows [Versión 10.0.16299.431]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Reggie Barker\Documents\TEC\Diököl>diokol.exe proyecto.lua
```

4- La siguiente ventana es el resultado final de la lectura de los archivos JSON o csv quienes son los que obtienen los datos para poder escribir el gráfico.



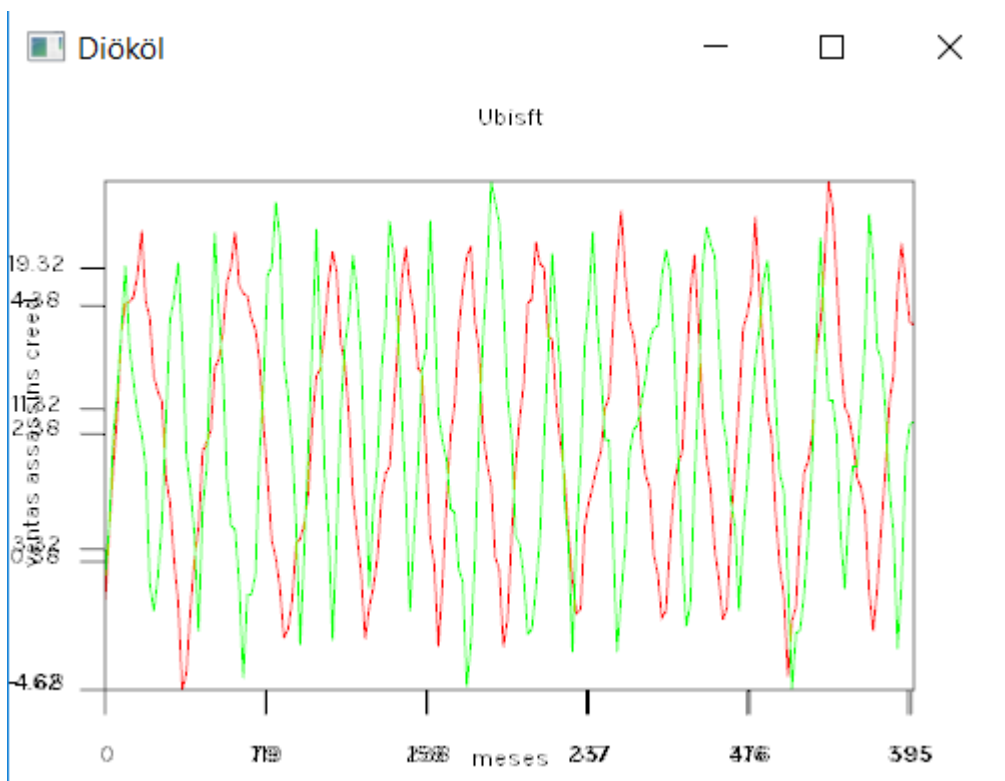
Ejemplo de la aplicación:

Ejecución:

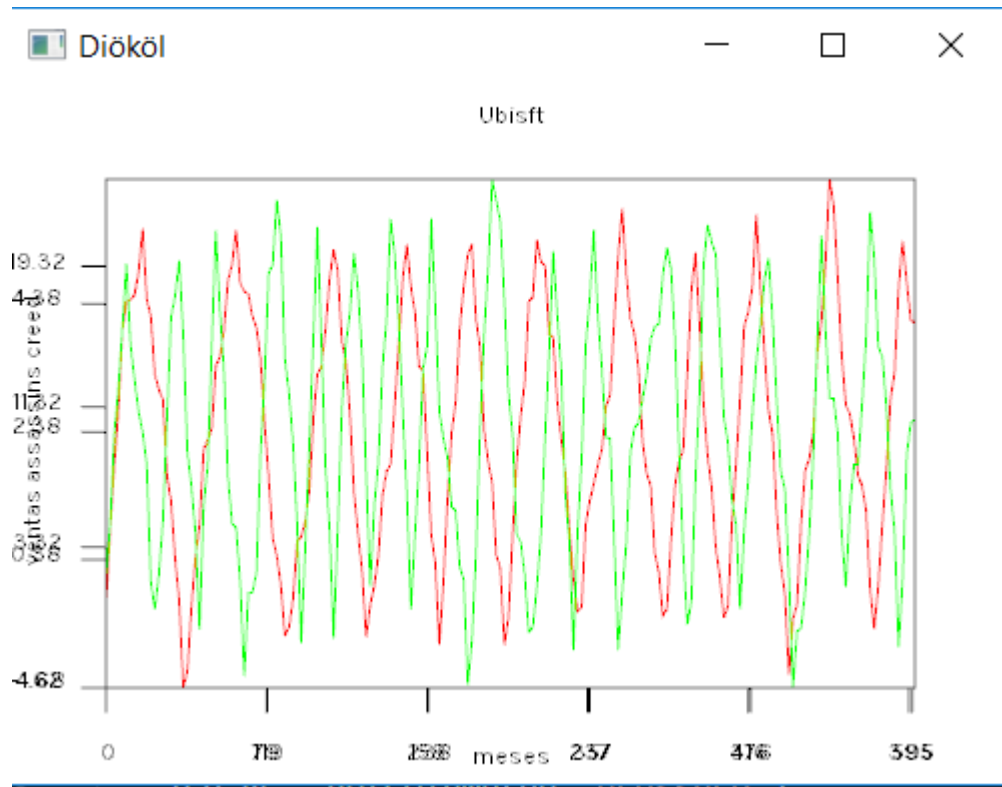
```
C:\Windows\System32\cmd.exe - diokol.exe proyecto.lua
Microsoft Windows [Versión 10.0.16299.431]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Reggie Barker\Documents\TEC\Diököl>diokol.exe proyecto.lua
^C
C:\Users\Reggie Barker\Documents\TEC\Diököl>diokol.exe proyecto.lua
```

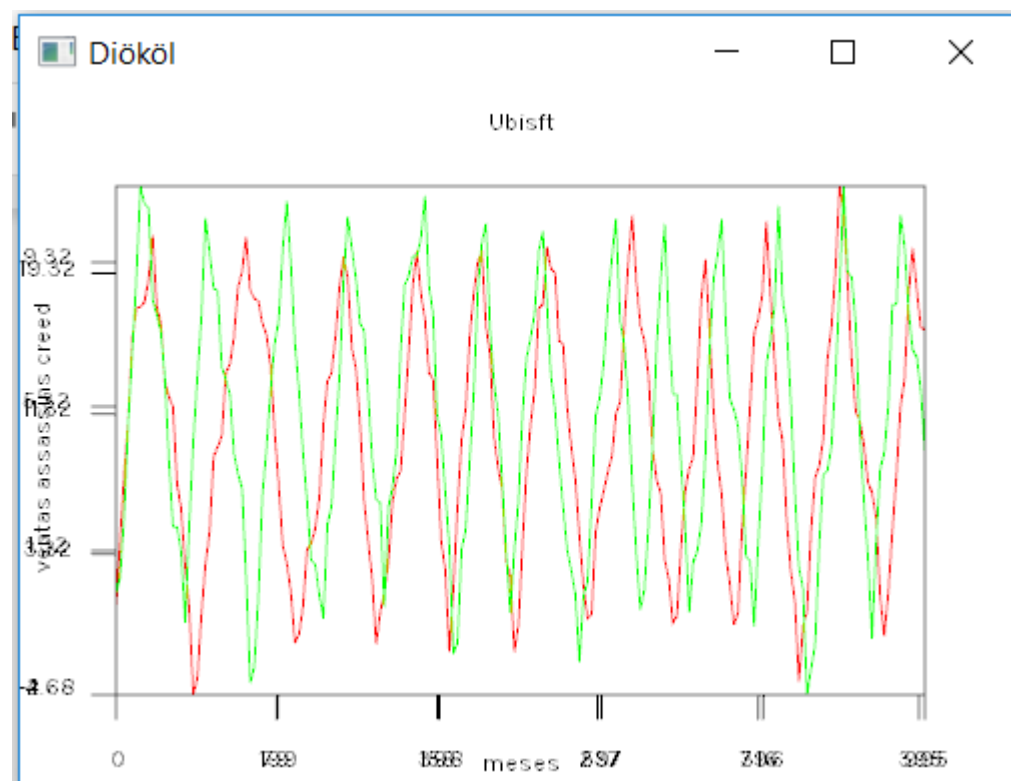
Resultado datos 1:



Resultado datos 2:



Resultado datos 3:



Referencias:

<https://flowingdata.com/2015/07/02/changing-price-of-food-items-and-horizon-graphs/>
<https://www.perceptualedge.com/blog/?p=390>
<https://dataveld.com/2017/04/04/building-horizon-graphs-in-power-bi/>
<http://vis.berkeley.edu/papers/horizon/2009-TimeSeries-CHI.pdf>

<http://mbostock.github.io/protovis/ex/horizon.html>