

AUTOMATIZACIÓN E IMPLEMENTACIÓN DEL CÓDIGO DE FILTROS DE KALMAN PARA EL AJUSTE DE LAS SALIDAS DEL MODELO GFS EN LAS VARIABLES PRECIPITACIÓN Y TEMPERATURAS MÍNIMA, MEDIA Y MÁXIMA

Jennifer Dorado*, Jeimmy Yanelly Melo** y José Franklyn Ruiz**

*Contratista Subdirección de Meteorología (Grupo de Modelamiento de Tiempo y Clima) - IDEAM, Bogotá, Colombia

** Profesional Subdirección de Meteorología (Grupo de Modelamiento de Tiempo y Clima) - IDEAM, Bogotá, Colombia
e-mail: jdoradod82@gmail.com

RESUMEN: Este artículo contiene la descripción de la automatización del código de Filtro de Kalman para la aplicabilidad en la subdirección de meteorología del IDEAM, que permitirá conocer el método de ajuste o corrección de los datos del modelo GFS de forma fácil, a partir de la optimización de las ecuaciones de actualización para las variables a analizar. Para este trabajo, se contó con los datos observados de 42 estaciones de los aeropuertos del IDEAM y la información del sistema de predicción GFS. El método de corrección se aplicó para las variables precipitación y temperaturas (mínima, media y máxima), a través de un código en bash y scripts de R. Se realizó evaluación por correlación, BIAS y MAE, para identificar la eficiencia del algoritmo. Los resultados obtenidos indicaron que el Filtro de Kalman como método de ajuste y/o corrección, mejora las salidas de las proyecciones del GFS para un horizonte de 7 días en las 4 variables analizadas.

PALABRAS CLAVES: *Filtro de Kalman, GFS, Precipitación, Temperatura máxima, Temperatura media, Temperatura mínima.*

ABSTRACT: *This article contains the description of the automation of the Kalman Filter code for the applicability in the IDEAM sub-direction of meteorology, which will allow to know the method of adjustment or correction of the GFS model data easily, from the optimization of the update equations for the variables to analyze. For this work, the observed data from 42 IDEAM airport stations and the GFS prediction system information were available. The correction method was applied for the precipitation and temperature variables (minimum, average and maximum), through a code in bash and R scripts. Correlation evaluation, BIAS and MAE, was performed to identify the efficiency of the algorithm. The results obtained indicated that the Kalman Filter as a method of adjustment and / or correction improves the outputs of the GFS projections for a 7-day horizon in the 4 variables analyzed..*

KEYWORDS: Kalman Filter, GFS,

1. INTRODUCCIÓN

El Filtro de Kalman, es un método que permite minimizar los ruidos generados en las series dinámicas de un sistema, por medio de un algoritmo. La ventaja de este respecto a otros, es que permite escoger la forma óptima de la serie, a partir de la identificación de las varianzas de los ruidos que afectan al sistema (Deossa, 2009).

La eficiencia del método se basa en que las ecuaciones solo dependen de un dato o muestra anterior (t-1) a la muestra presente, lo que permite optimizar los procesos con bajo recurso de memoria serial para su implementación.

Uno de los retos que enfrenta la sistematización del Filtro de Kalman, es la aplicabilidad del método en sistemas no lineales como es el caso de la precipitación; para ello, se trabaja una extensión del filtro con un proceso de linealización que se conoce como Filtro de Kalman Extendido (FKE), (Haykin, 1999), donde linealiza el modelo de estado no lineal.

En cuanto a la aplicabilidad del Filtro en los modelos meteorológicos de predicción, para este caso sistema de predicción GFS-FV3, este permite mejorar o minimizar los errores sistemáticos que pueden ser derivados, por ejemplo, en parametrizaciones físicas (elevación del lugar, vegetación, etc.) y en los datos que inicializan al modelo. Por esta razón, muchos

centros meteorológicos utilizan métodos de post-procesamiento tratando de filtrar ruidos y errores sistemáticos implementando algún tipo de método adaptivo de regresión dinámica (Chacón 2009).

Por su practicidad y eficiencia, se realiza la automatización e implementación del método de Filtro de Kalman, con el fin de mejorar las salidas del modelo GFS-FV3 que le permita al IDEAM, mejorar la calidad de las series proyectadas del pronóstico del tiempo en las variables anteriormente mencionadas.

2. MATERIALES Y MÉTODOS

Aunque existen automatizaciones del Filtro de Kalman que resumen la eficiencia del método, para la aplicabilidad del Filtro en el IDEAM, se requirió diseñar códigos para la lectura de los archivos Excel correspondientes a los datos observados, que se encuentran publicados en la intranet del IDEAM, debido a que los formatos no son unificados (archivos diferentes para cada variable).

Los datos de entrada son los datos observados de 42 estaciones convencionales meteorológicas del IDEAM (figura 1) y los datos históricos del modelo del GFS (t-1) para los 42 puntos analizados.

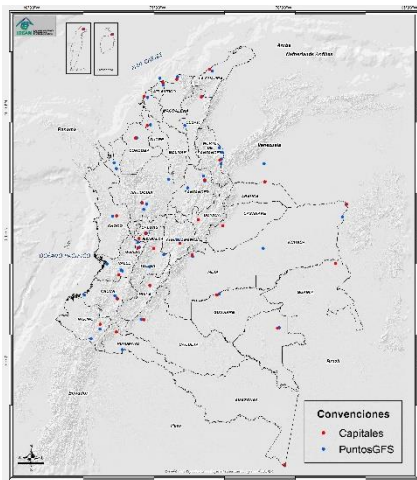


Figura 1. Distribución espacial de los Aeropuertos evaluados

Para obtener los datos de la intranet para las 4 variables, se realizaron 2 script en R, uno para temperatura (generarDatosObsTemps-v1.R) y otro para precipitación (generarDatosObsPrec-v1.R), permitiendo así extraer y organizar la información de la intranet al formato de entradas del Filtro de Kalman. Es importante mencionar que ambos archivos tienen diferente estructura cuando se

descargan de intranet. En cuanto a la información a descargar del GFS FV3 en formato GRIB. Se emplearon los script `getdataGFS.gs` para extraer las series de los puntos de los aeropuertos a partir de los datos del GFS y el `g2ctl.pl` del link <http://www.ftp.ncep.noaa.gov/data/nccf/com/gfs/pr od/gfs>, en lenguaje Perl, a fin de generar tablas para la visualización y análisis de datos climáticos en GrADS.

El funcionamiento del Filtro de Kalman, se basa en la optimización sucesiva del coeficiente de actualización de las salidas del GFS para las temperaturas y la precipitación.

Los aspectos metodológicos fueron basados en la aplicabilidad del Filtro de Kalman como método de ajuste (corrección) de las salidas a partir de la optimización sucesiva del coeficiente de actualización.

Para diseñar el Filtro de Kalman, se debe analizar el tipo de variable y/o proceso a realizar y al igual que determinar el ruido de las medidas ya que este filtro tiene un diseño particular y es el de obtener una respuesta rápida y sensible frente a las perturbaciones.

El diseño propuesto original del Filtro corresponde a sistemas lineales como es el caso de las temperaturas y Filtro de Kalman Extendido y el Unscented para los sistemas no lineal como es la precipitación.

El método del Filtro de Kalman Extendido permite llevar a lineal un sistema no lineal y el Unscented, permite reducir los errores introducidos por la linealización que realiza el FKE. El Filtro Unscented, permite aproximar la densidad de la probabilidad a partir de su propia media y covarianza y fue el método aplicado en este trabajo.

El método basado en la linealidad para una variable discreta, original de la propuesta del Filtro de Kalman, se basa en indicar que, aunque el valor del coeficiente cierto C_1^A , es desconocido, el coeficiente de presumible en ese punto, C_t , se puede encontrar. El Filtro de Kalman es un método para encontrarlo mediante la utilización de las observaciones de series de tiempo (Y_0, Y_1, \dots, Y_t) para actualizar los coeficientes de C_t en diferentes puntos de tiempo a coeficientes más presumibles.

El coeficiente de actualización (optimización sucesiva), se realizó teniendo en cuenta que en el tiempo t , el coeficiente presumible en el momento

time $t-1$, C_{t-1} , es conocida y la variable explicativa en el momento t , X_t , se ha calculado a partir de la predicción numérica del tiempo en $t-1$. Entonces, las ecuaciones (1) y (2) pueden ser utilizados para obtener la primera estimación de C_t , C^-_t , y la estimación de Y_t , Y^-_t , tal como se describe a continuación:

$$Y_t = C^*_t X_t + v_t \quad \text{Relación entre los coeficientes y observaciones} \quad (1)$$

$$C^*_t = A + C^-_{t-1} + u_{t-1} \quad \text{Relación de coeficientes con coeficientes en el punto de tiempo anterior} \quad (2)$$

En primer lugar, calcula la primera estimación de C_t , C^-_t , de la ecuación (2). Debido u_t es desconocido en la ecuación, ignorar la cantidad que se muestra a continuación. Para obtener orientación, se puede asumir que los coeficientes no varían sistemáticamente con el tiempo y que $A = I$ (unidad de matriz).

$$C^-_t = C^-_{t-1} \quad (3)$$

Reemplace C_t en la ecuación (1) con la primera estimación de C_t , C^-_t , y calcular Y^-_t estimación como se muestra a continuación. Ignorar error v_t .

$$Y^-_t = C^-_t X_t \quad (4)$$

C_{t-1} : Coeficiente de presumible calculado a partir de observaciones en el momento $t-1$

C^-_t : Primera estimación del coeficiente en el tiempo t calcula a partir de coeficiente presumible C_{t-1}

Y^-_t : Estimación de Y_t en el tiempo t calculado desde la primera estimación del coeficiente C^-_t

A continuación, corregir la primera estimación C^-_t de acuerdo a la diferencia entre la observación Y_t , para obtener un coeficiente más presumible C^-_t .

El filtro de Kalman está diseñado originalmente para actualizar un coeficiente sucesivamente a un coeficiente más presumible C_t .

C^-_t se actualiza como se muestra abajo. Esta es la ecuación de Kalman.

$$C_t = C^-_t + \partial_t (Y_t - Y^-_t) \quad (5)$$

∂_t : Velocidad a la que se cambian los coeficientes (ganancia de Kalman)

Mediante la repetición de las ecuaciones (3), (4), y (5), C_t debe actualizarse sucesivamente. El punto es como decir ∂_t .

Generalmente, las ecuaciones de predicción contienen dos o más coeficientes. Sin embargo, para

simplificar el proceso, la siguiente descripción utiliza las ecuaciones unidimensionales.

Aunque C_t es el coeficiente más presumible en ese punto, que tiene un cierto error con respecto al verdadero valor de C^-_t . Definir el error como e_t .

$C^*_t = C_t + e_t$: Error e_t es el ruido blanco.

El error es expresado en las siguientes ecuaciones:

$$e_t = C^*_t - C_t \quad \text{Ecuación se suple en (5)}$$

$$e_t = C^*_t - (C^-_t + \partial_t (Y_t - Y^-_t)) \quad \text{Sustituye (1) y (4)}$$

$$e_t = C^*_t - (C^-_t + \partial_t (C^*_t X_t + v_t - C^-_t X_t))$$

$$e_t = (1 - X_t \partial_t) (C^*_t - C^-_t) - \partial_t v_t$$

$$e_t = (1 - X_t \partial_t) (C^*_t - C^-_t) - \partial_t v_t$$

$$e_t = (1 - X_t \partial_t) (C^*_{t-1} + u_{t-1} - C^-_{t-1}) - \partial_t v_t$$

$$e_t = (1 - X_t \partial_t) (u_{t-1} - e_{t-1}) - \partial_t v_t \quad (6)$$

Definir varianza Q_t en ecuación (6) de la siguiente manera:

$Q_t = \text{Var} (e_t)$; (Varianza de error del valor del coeficiente presumible).

La ecuación (6) puede ser utilizado para desarrollar Q_t de la siguiente manera:

$$Q_t = \frac{1}{n} \sum (e_t)^2$$

$$Q_t = (1 - X_t \partial_t)^2 (U_{t-1} + Q_{t-1}) + \partial_t^2 D_t$$

$$Q_t = \partial_t^2 (X_t^2 (U_{t-1} + Q_{t-1}) + D_t) - 2 \partial_t X_t (U_{t-1} + Q_{t-1}) + (U_{t-1} + Q_{t-1}) \quad (7)$$

Q_t es una ecuación bidimensional de δ_t . El δ_t que minimiza Q_t se puede obtener cuando el diferencial parcial debido a δ_t es 0.

$$2 \partial_t (X_t^2 (U_{t-1} + Q_{t-1}) + D_t) - 2 X_t (U_{t-1} + Q_{t-1}) = 0$$

$$\partial_t = X_t (U_{t-1} + Q_{t-1}) / (X_t^2 (U_{t-1} + Q_{t-1}) + D_t) \quad (8)$$

Suponiendo que Q_{t-1} es conocido, δ_t puede ser calculado a partir de la ecuación (8).

El Q_t se puede calcular mediante la sustitución de la ecuación (8) en la ecuación (7).

$$Q_t = (U_{t-1} + Q_{t-1}) - X_t^2 (U_{t-1} + Q_{t-1})^2 / (X_t^2 (U_{t-1} + Q_{t-1}) + D_t)$$

$$Q_t = D_t (U_{t-1} + Q_{t-1}) / (X_t^2 (U_{t-1} + Q_{t-1}) + D_t) \quad (9)$$

La identificación de la variable explicativa se tomó a partir encontrar aquella que tiene peso sobre el cambio de la variable de respuesta. Para este estudio, se tomó en cuenta que:

D_t : Es la varianza del error de observación (Una gran varianza hace que sea difícil para los coeficientes que cambian.).

Q_o : Coeficiente de varianza del error inicial

U_t : varianza de error del sistema (A gran diferencia hace que sea fácil para que los coeficientes cambien)

X0/0: Los valores iniciales de coeficientes

Una vez identificada la metodología del Filtro de Kalman, se procedió a generar la estructura que compone la automatización e implementación de este método en las variables de salidas del GFS, para ello, a continuación, se realiza una breve descripción de cada uno de los script realizados y la funcionalidad de ellos, partiendo con ejecutable correrFiltroKalman.bash, donde se reúnen cada uno de los scripts.

2.1 Script correrFiltroKalman.bash

La generación de este script en bash, se construyó con el objetivo de compilar en un solo ejecutable, todo el proceso de descarga de los datos del GFS e intranet para el posterior funcionamiento del Filtro de Kalman. Es importante indicar que el Filtro de Kalman original, se encuentra estructurado en R para Linux y fue creado en el año 2013 en el aplicativo llamado ApplyKalmanFilter_Linux con entradas manuales para estaciones individuales. De acuerdo a la importancia y los resultados obtenidos dentro con el Filtro en el año 2013, se adecuó en un 100% automatización de las corridas seriales con información propia del IDEAM para su funcionamiento dentro de la institución.

Para que la corrida del script se genere con éxito, es importante definir la ruta de la ubicación de la carpeta de trabajo (fila 5), es decir, lo que se considerará carpetadatos.

```
#5 carpetadatos="/home/ideam/Filtro_kalman"
```

Entre las filas 12 y 14, se ubica de forma automática la fecha del año, mes y día de la descarga en curso. Es importante no modificar estos parámetros.

```
#12 export iniyear=`date +%Y`
```

```
#13 export inimonth=`date +%m`
```

```
#14 export iniday=`date +%d`
```

En caso de querer descargar una fecha del GFS diferente al día actual, se pueden inhabilitar las filas 12 al 14 con # y fije la fecha a descargar en las filas 15 al 17.

```
#15 export iniyear=2019
```

```
#16 export inimonth=10
```

```
#17 export iniday=17
```

La resolución de los datos del GFS para descargar, se indica en la fila 19. Recuerde que las opciones corresponden a 0.25, 0.50, 1.00. Para el caso del script, se cambia el punto (.) por la p.

```
#19 resolucion="0p25"
```

Entre las filas 35 y 82 se encuentran los comandos de creación de las carpetas para los datos del GFS, las salidas Kalman y las subcarpetas de series, estadísticos y gráficas para cada variable. Es importante NO MOVER NINGUNO DE LOS CONTENIDOS DE ESTAS FILAS.

La estructura del Filtro de Kalman, contiene 4 carpetas llamadas: scripts (todos los ejecutables en R o bash), salidasKalman (gráficos, series y estadísticos por variable), instaladores (los paquetes necesarios para futuras instalaciones) y datos_entradas (se ubica intranet, datosOBS, datosGFS y puntos.txt)

De la fila 84 a la 106, se encuentran los comandos correspondientes al paso 1 llamado *Descarga de los datos del GFS*. El usuario solo debe tener en cuenta modificar la línea 90 donde debe indicar el intervalo de descarga. Las opciones son (00, 06, 12 y 18). Para este caso, la descarga se hace cada 6 horas.

```
#90 export inihour=6
```

Tenga en cuenta que este aplicativo está diseñado para un horizonte de 7 días por lo que las horas de descarga corresponden hasta 168 horas (24horas/día * 7 días de horizonte). De acuerdo a esto, en la fila 94 se indica que descargue las horas menores a 169

```
#94 while [ $inihour -lt 169 ]; do
```

La estructura de los datos del GFS en las horas, se encuentra estructurado para 3 dígitos f000 hasta f168, por lo cual entre las filas 94 y 100, se realiza un ajuste para cumplir con la estructura inicial. Es por esto que encuentran -lt 10 (fila 95) para f00 y -lt 100 (fila 98) para f0. Estos parámetros NO DEBEN SER MODIFICADOS POR EL USUARIO.

En el paso 2, *Conversión a GrADS los archivos crudos del GFS*, se encuentran los comandos respectivos para obtener las series de datos correspondientes a cada estación. Es importante

tener instalado los programas GrADS, gribmap y wgrib2.

Para que se genere los archivos GrADS, es importante tener los archivos ctl e idx, por lo cual entre las líneas 117 y 122 se realiza este proceso. El usuario NO DEBE MODIFICAR ESTOS PARÁMETROS.

```
#117 echo "Generando ctl e idx para el archivo
$FILE" #114
${carpetadatos}/scripts/.g2ctl.pl -0 $FILE >
$FILE.ctl (EN EL PUNTO 2.2 Se explicará más
sobre el script g2ctl.pl)
#119 gribmap -0 -i $FILE.ctl

#120 done
#121
#122 echo "Conversión a GrADS finalizada"
```

En el paso 3. *Se generan las series de datos del GFS* para cada estación a partir de los datos descargados y convertidos a formato GrADS, por lo cual se hace lectura de los puntos, para este caso de los aeropuertos. El archivo que contiene las coordenadas y descripción de los puntos a analizar, se encuentran en el archivo *puntos.txt*, ubicado dentro de la carpeta de datos_entradas. El usuario NO DEBE MODIFICAR NINGUNA DE LAS RUTAS NI PARÁMETROS entre las líneas 133 a 140. EN EL PUNTO 2.3 Se explicará más sobre el script getdataGFS.gs).

Finalmente, en el punto 4, se corre el programa principal del Filtro de Kalman, para ello se ejecuta en las líneas 146 la descarga de los datos de la intranet para temperatura, 147 descarga de datos de la intranet para la precipitación, 148 el script del Filtro de Kalman versión 2019, 149 el script para mapas los de precipitación y 150 el script para los mapas de temperaturas

```
#146 Rscript ${carpetadatos}/scripts/generarDatosObsTemps-v1.R
${iniyear} ${inimonth} ${iniday} ${carpetadatos}

#147 Rscript ${carpetadatos}/scripts/generarDatosObsPrec-v1.R
${iniyear} ${inimonth} ${iniday} ${carpetadatos}

#148 Rscript ${carpetadatos}/scripts/ApplyKalmanFilter_v2019.r
${iniyear} ${inimonth} ${iniday} ${horagfs} ${carpetadatos}

#149 Rscript ${carpetadatos}/scripts/generarMapasKalmanPrec-v1.R
${iniyear} ${inimonth} ${iniday} ${horagfs} ${carpetadatos}

#150 Rscript ${carpetadatos}/scripts/generarMapasKalmanTemps-
v1.R ${iniyear} ${inimonth} ${iniday} ${horagfs} ${carpetadatos}

#151 Fin
```

Estas rutas son las que se deberán seguir en caso de querer correr de forma individual el script desde la terminal de R. Recuerde que carpetadatos es la ruta indicada en la fila 5.

2.2 Script g2ctl.pl

Este script está en lenguaje Perl y permite realizar la decodificación de los archivos del GFS que vienen en formato GRIB, generando tablas para utilizar estos archivos en la herramienta de visualización y análisis de datos climáticos GrADS. La estructura del mismo, es estático y se descarga de la página <https://www.cpc.ncep.noaa.gov/products/wesley/g2ctl.html>. El usuario NO DEBE MODIFICAR NINGUNO DE LOS PARÁMETROS, solo en la Fila 42, se deberá indicar la ruta de ubicación del archivo wgrib2.

```
#42$wgrib2='/home/ideam/Filtro_kalman/instala
dores/grib2/wgrib2/wgrib2';
```

La descarga del wgrib2, la puede obtener en la página <https://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/index.html>. Sin embargo, dentro de la carpeta principal del Filtro, se tiene dentro de instaladores, todos los ejecutables necesarios para el funcionamiento del mismo.

2.3 Script getdataGFS.gs

Este script permite extraer las series del GFS para los puntos indicados a analizar, para este caso, los aeropuertos, de las variables precipitación, temperatura mínima, media y máxima. El usuario NO DEBE MODIFICAR NINGUNO DE LOS PARÁMETROS.

2.4 Script generarDatosObsTemps-v1.R

Este script está diseñado para la descargar y organizar los datos de la intranet de las variables temperatura media, mínima y máxima.

Es importante mencionar que la estructura en que se suban los datos a la intranet, debe ser constantes a lo subido hasta octubre del 2019, en caso de modificar la tabla desde su origen, el filtro no contará con el dato observado, por lo cual es importante que desde la Oficina de Pronóstico, se mantenga el formato.

El script está en lenguaje R y para su funcionamiento debe verificar que se cuente con las librerías downloader, readxl, dplyr, tidyr, openxlsx y lubridate.

```
#8 library(downloader)
#9 library(readxl)
#10 library(dplyr)
#11 library(tidyr)
#12 library(openxlsx)
#13 library(lubridate)
```

En las filas 15 a la 18, está la estructura para la fecha de descarga (año, mes día), este se hará de acuerdo al día actual de ejecución.

```
#15 anyo=commandArgs(TRUE)[1]
#16 mes=commandArgs(TRUE)[2]
#17 dia=commandArgs(TRUE)[3]
#18 carpetadatos=commandArgs(TRUE)[4]
```

En caso de querer descargar días diferentes al actual, deberá inhabilitar las filas 15-18 y habilitar las del 20-22, indicando el año, mes (para meses menores a 10, colocar entre “” el dato) y día.

```
#20 anyo=2019
#21 mes="09"
#22 dia=18
#23 carpetadatos="/home/ideam/Filtro_kalman"
```

No se preocupe por la creación de las carpetas, estas las genera el programa automáticamente.

La descarga se realiza en las filas 44-49 para cada una de las variables. Es importante tener en cuenta que esta solo se realiza dentro del IDEAM ya que depende de la dirección de intranet de libre acceso.

```
#44 link21 =
"http://sindamanoy.ideam.gov.co:8080/intranetv2/Ftpasp/Servici
opt/Temperaturaminima.xls"
#45 download(link21,destfile =
paste0(dir,"Temperaturaminima.xls"),mode = "wb")
#46 link22 =
"http://sindamanoy.ideam.gov.co:8080/intranetv2/Ftpasp/Servici
opt/Temperaturamed.xls"
#47 download(link22,destfile =
paste0(dir,"Temperaturamed.xls"),mode = "wb")
#48 link23 =
"http://sindamanoy.ideam.gov.co:8080/intranetv2/Ftpasp/Servici
opt/Temperaturamax.xls"
#49 download(link23,destfile =
paste0(dir,"Temperaturamax.xls"),mode = "wb")
```

Las filas siguientes, corresponden a la organización de los datos para que el Filtro de Kalman los pueda

leer, por lo cual el usuario NO DEBERÁ MODIFICAR LOS PARÁMETROS.

2.5 Script generarDatosObsPrec-v1.R

Este script está diseñado para la descargar y organizar los datos de la intranet para la variable precipitación.

Es importante mencionar que la estructura en que se suban los datos a la intranet, debe ser constantes a lo subido hasta octubre del 2019, en caso de modificar la tabla desde su origen, el filtro no contará con el dato observado, por lo cual es importante que desde la Oficina de Pronóstico, se mantenga el formato.

El script está en lenguaje R y para su funcionamiento debe verificar que se cuente con las librerías downloader, readxl, dplyr, tidyr, openxlsx y lubridate.

```
3. #8 library(downloader)
4. #9 library(readxl)
5. #10 library(dplyr)
6. #11 library(tidyr)
7. #12 library(openxlsx)
8. #13 library(lubridate)
```

En las filas 15 a la 18, está la estructura para la fecha de descarga (año, mes día), este se hará de acuerdo al día actual de ejecución.

```
#15 anyo=commandArgs(TRUE)[1]
#16 mes=commandArgs(TRUE)[2]
#17 dia=commandArgs(TRUE)[3]
#18 carpetadatos=commandArgs(TRUE)[4]
```

En caso de querer descargar días diferentes al actual, deberá inhabilitar las filas 15-18 y habilitar las del 20-22, indicando el año, mes (para meses menores a 10, colocar entre “” el dato) y día.

```
#20 anyo=2019
#21 mes="09"
#22 dia=18
#23 carpetadatos="/home/ideam/Filtro_kalman"
```

No se preocupe por la creación de las carpetas, estas las genera el programa automáticamente.

La descarga se realiza en las filas 39-40 para cada una de las variables. Es importante tener en cuenta

que esta solo se realiza dentro del IDEAM ya que depende de la dirección de intranet de libre acceso.

```
#39 link24 =  
http://sindamanoy.ideam.gov.co:8080/intranetv2/Ftpasp/Servici  
opt/Precipitacion.xls"
```

```
# 40 download(link24,destfile =  
paste0(dir,"Precipitacion.xls"),mode = "wb")
```

Las filas siguientes, corresponden a la organización de los datos para que el Filtro de Kalman los pueda leer, por lo cual el usuario NO DEBERÁ MODIFICAR LOS PARÁMETROS.

2.6 Script ApplyKalmanFilter_v2019.r

Este es el script que contiene toda la funcionalidad de Filtro de Kalman. Inicialmente, identifica la fecha actual para ejecutar el algoritmo. En las filas 7-10 se ubican las rutas de fecha, año, mes, día y horagfs. Adicionalmente en la fila 11, se indica la ubicación de la carpeta de datos.

```
#7 anyo=commandArgs(TRUE)[1]  
#8 mes=commandArgs(TRUE)[2]  
#9 dia=commandArgs(TRUE)[3]  
#10 horagfs=commandArgs(TRUE)[4]  
#11 carpetadatos=commandArgs(TRUE)[5]
```

En caso de querer aplicar el Filtro para un día diferente al actual, deberá inhabilitar las filas 7-10 y habilitar las del 13-16, indicando el año, mes (para meses menores a 10, colocar entre "" el dato) y día.

```
#13 anyo=2019  
#14 mes="09"  
#15 dia=18
```

En la fila 20, se lee el archivo de coordenadas de puntos.

```
#20estaciones0=read.table(paste(carpetadatos, "/datos_en  
tradas/puntos.txt", sep = ""),header=FALSE, dec = ",",  
sep="\t")
```

De la línea de comandos 28-35 son parámetros propios del filtro¹. Sin embargo, la línea de comando

#28, es ajustable al valor de A que defina según su criterio (Recuerde: A es la matriz para estimar la cantidad del estado para el siguiente punto de tiempo). Para el caso donde la amplitud de los datos observados vs los modelados por el GFS sea demasiado grande, use (-) si el GFS está por debajo del observado y (+) si el GFS está por encima del observado, ambos según lo representado gráficamente. Por defecto el programa usará A_{proy}=1. Si desea puede usar para el caso de temperaturas el A=0, solo tenga presente que así mismo será el comportamiento de la serie proyectada, es decir, con un A=0 tenderá más hacia la neutralidad y un A=1 tendrá una mejor amplitud del observado histórico.

```
#28 A=3  
#29 A_proy=1  
#30 q_f=1  
#31 r_f=0.1  
#32 q_f0=q_f/3  
#33 q_f00=q_f/2
```

En la fila 35, se debe indicar el horizonte (días) a proyectar. Para este estudio, se trabaja con horizonte a 7 días.

```
#35 M = 7
```

En la fila 40, se ubica la carpeta de los datos del GFS pronosticados que se encuentran dentro de los datos_entradas en la subcarpeta datosGFS. Cada carpeta se encuentra con el nombre GFS año, mes, día y hora de la salidaGFS.

```
#40carpetadatosGFS=paste(carpetadatos, "/datos_en  
tradas/datosGFS/GFS",anyo,mes,dia,horagfs, "/serie  
s",sep="")
```

En la fila 43, se ubica la carpeta de los datos históricos observados y del GFS histórico. Es importante indicar que realizar el ajuste y/o corrección de los datos, se debe tener el mismo número de días base, es decir, si se va a proyectar el 2 de octubre, se debe tener una base completa y nivelada del 1 de octubre tanto de observado de los datos de la intranet como el del GFS histórico,

¹ Dorado, J; Ruíz, J. F. IMPLEMENTACIÓN DE FILTROS DE KALMAN COMO MÉTODO DE AJUSTE A LOS MODELOS DE PRONÓSTICO (GFS) DE TEMPERATURAS MÁXIMAS Y MÍNIMA PARA

siendo para este último, el día 1 de la proyección del día anterior (para nuestro ejemplo, el 1 de octubre). Los nombres de los archivos que conforman la carpeta de observados, es el código de la estación con -GFS ó -Obs según el caso.

```
#43carpetadatosObs=paste(carpetadatos,"/datos_en
tradas/datosOBS",sep="")
```

Las salidas del Filtro de Kalman, se encuentran en la carpeta salidasKalman.

```
#carpetasalidas=paste(carpetadatos,"salidasKalma
n",sep="")
```

Para conocer la veracidad de las salidas del ajuste del Filtro, se generan 3 estadísticos que son correlación, BIAS y MAE y se hace para cada variable del observado de la intranet con el GFS original y el GFS corregido llamado GFS-K. La generación de la tabla de estos estadísticos, se generan entre las filas 50 y 77 para las 4 variables.

El nombre del archivo de salida de los estadísticos se hace en la fila 80

```
#80archivoSalidaEstadisticos=paste("estadisticos/E
stadisticos_GFS",anyo,mes,dia,"_",horagfs,"Z.txt",
sep="")
```

El procesamiento del Filtro de Kalman inicia a partir de la fila 83, partiendo en la generación de los nombres de los archivos históricos tanto del observado de la intranet como del GFS, las salidas del filtro y los nombres de la estación y fecha del GFS que se procesa en las filas 89-100.

El cálculo de los valores diarios proyectados a horizonte de 7 días, se realiza a partir de la lectura de los archivos de los datos del GFS procesados y extraídos previamente. Este paso se realiza en las filas 105-108.

La lectura de los datos del GFS, desde la extracción del dato, codificación y nombre, organización del dato en columnas y almacenamiento de la información, se realiza entre las filas 116-143.

Una vez obtenidos los datos base para el filtro, correspondiente a los observados de la intranet y del GFS, se inicia con abrirlos.

Datos Observados

```
#148 archObs=paste(carpetadatosObs, "/", varmet, "/",
archivoObs, sep = "")
#149datosObs=read.delim(archObs,header=TRUE,sep="
",row.names=NULL)
```

Datos GFS

```
#151 archGFS=paste(carpetadatosObs, "/", varmet, "/",
archivoGFS, sep = "")
#152datosGFS=read.delim(archGFS,header=TRUE,row.
names=NULL)
```

Para que el Filtro funcione, la cantidad de datos históricos deben ser igual en ambas series (observados de la intranet y del GFS), este proceso se realiza con:

```
#155 N = nrow(datosObs)
```

Una vez se identifica que todos los parámetros de entrada están debidamente organizados, se procede a realizar los cálculos respectivos del Filtro a partir de la fila 159. Es importante indicar que el usuario **NO DEBE MODIFICAR NINGÚN PARÁMETRO**.

De acuerdo al esquema metodológico del Filtro mencionado inicialmente, las parametrizaciones dentro del script fueron:

```
#159 Dt=sd(datosObs[,5], na.rm=TRUE)
```

```
#161 varianza <- array(9999,dim=c(1,N))
for (i in 1:N){
  varianza[i]=var(c(datosObs[i,5],datosGFS[i,5]),
na.rm=TRUE)
}
```

```
#166 Ut <- array(9999,dim=c(1,N))
for (i in 1:N){
  Ut[i]=(1/N)*varianza[i]
}
```

```
#171 Ct <- array(9999,dim=c(1,N))
for (i in 1:N){
  if((datosGFS[i,5]==0) | (is.na(datosObs[i,5]))){
    Ct[i]=1
  }else{
    Ct[i]=datosObs[i,5]/datosGFS[i,5]
  }
}
```

```
#180 Utant <- array(9999,dim=c(1,(N-1)))
for (i in 2:N){
  if(Ct[i-1]==0){
    Utant[i-1]=1
  }else{
    Utant[i-1]=Ct[i]/Ct[i-1]
  }
}
```



```

#189 C_t <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){
  C_t[i]=(A*Ct[i])+Utant[i]
}

#194 Vt <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){
  if((datosGFS[(i+1),5]==0) | ((Ct[i]==0) |
(is.na(datosObs[(i+1),5])))){
    Vt[i]=1
  }else{
    Vt[i]=datosObs[(i+1),5]/(C_t[i]*datosGFS[(i+1),5])
  }
}

#203 ECMR <- array(9999,dim=c(1,N))
for (i in 1:N){
  if(is.na(datosObs[i,5])){
    ECMR[i]=0
  }else{
    ECMR[i]=(datosObs[i,5]-datosGFS[i,5])^2
  }
}
#211 UtConst=(sum(ECMR)/N)^(1/2)

#213 Ec1Treal <- array(9999,dim=c(1,(N+M)))
for (i in 1:(N-1)){
  Ec1Treal[i]=(Ct[i+1]*datosGFS[(i+1),5])+Vt[i]
}

#218 Ec4 <- array(9999,dim=c(1,N))
for (i in 1:N){
  if(is.na(datosObs[i,5])){
    Ec4[i]=0
  }else{
    Ec4[i]=datosObs[i,5]*datosGFS[i,5]
  }
}

#227 Ec8 <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){
  if(datosGFS[(i+1),5]==0){
    Ec8[i]=1
  }else{
    Ec8[i]=(datosGFS[(i+1),5]*(Utant[i]+varianza[i]))/(((dat
osGFS[(i+1),5])^2)*(Utant[i]+varianza[i])+Dt)
  }
}

#237 Ec5 <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){
  # if(is.na(datosObs[i+1,5])){
  # Ec5[i]=0
  # }else{
  Ec5[i]=datosObs[i+1,5]+(Ec8[i]*(Ec1Treal[i]-Ec4[i+1]))
  # }
}

#246 Ec9 <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){

```

```

  if(datosGFS[(i+1),5]==0){
    Ec9[i]=1
  }else{
    Ec9[i]=(Dt*(Utant[i]+varianza[i]))/(((datosGFS[(i+1),5])
^2)*(Utant[i]+varianza[i])+Dt)
  }
}

#255 C_t_corr <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){
  if(datosGFS[(i+1),5]==0){
    C_t_corr[i]=1
  }else{
    C_t_corr[i]=Ec1Treal[i]/datosGFS[(i+1),5]
  }
}
prom_C_t_corr=mean(C_t_corr,na.rm = TRUE)

#265 C_t_corr_fin <- array(9999,dim=c(1,M))
for (i in 1:M){
  C_t_corr_fin[i]=mean(C_t_corr, na.rm=TRUE)
}

#270 Ec1Treal_corr <- array(9999,dim=c(1,(N-1)))
for (i in 1:(N-1)){
  Ec1Treal_corr[i]=(C_t_corr[i]*Ec1Treal[i])+Dt
}

#275 Ec1Treal_corr_proy <- array(9999,dim=c(1,M))
for (i in 1:M){
  Ec1Treal_corr_proy[i]=(prom_C_t_corr*datosproy[i])+
A_proy
}

Entre las filas 285-321, se agregan a la ecuación 1
los valores supuestos calculados en la ecuación 1 de
la corrección del valor proyectado.

Una vez obtenido el resultado del Filtro de Kalman,
se procede a crear las tablas de las series de los datos
observados, el GFS original y el GFS corregido,
llamado GFS-K, para observar los resultados
obtenidos aplicando el método.

#325 titulosdatos<-c("Obs","GFS","GFS-K")
#326 datosfinales<-matrix(9999,nrow=(N+M-1),ncol=3)
#327 colnames(datosfinales)<-titulosdatos

Entre las filas 330-338, se colocan los resultados
obtenidos de este proceso.

#330 for (i in 1:(N+M-1)){
#331 datosfinales[i,1]=datosObs[i+1,5]
#332 if(i >= N){
#333 datosfinales[i,2]=datosproy[(i-(N-1))]
#334 }else{

```

```
#335 datosfinales[i,2]=datosGFS[i+1,5]
#336 }
#337 datosfinales[i,3]=GFS_K[i+1]
#338 }
```

Los estadísticos calculados correspondieron a: Correlación entre los observados del GFS y los GFS ajustados (filas 343-344). El BIAS entre 346-349 el MAE desde 352-355.

Para los gráficos, se crean los títulos entre las filas 358-363 y el gráfico de los datos entre las filas 366-386. Posteriormente, se escriben los datos de los estadísticos en la tabla de cada variable. Este proceso se evidencia en las filas 388-409.

En la fila 413 se realiza un proceso de redondeo de los valores de las series de datos observados, GFS y salidas del Filtro.

```
#413 datosfinales=round(datosfinales,1)
```

Entre las filas 416 a la 436, se colocan las fechas a la tabla final de los datos.

Desde la fila 439 hasta 441, se escriben las series de datos en el archivo de salida correspondiente.

```
#439 archsalidas=paste(carpetasalidas, "/", varmet,
"/", archivoSalida, sep = "")
#440 write.table(datosfinales, file = archsalidas,
sep = "\t",
#441 row.names = F, dec = ",", quote =
FALSE, qmethod = "double")
```

Finalmente, se actualizan los datos del GFS colocando el día reciente. Este se realiza entre las filas 445-460.

Fin del ejecutable.

Los resultados se generan en la carpeta salidasKalman en subcarpetas por variable. Cada variable tiene salidas gráficos, estadísticos, mapas y series.

2.7 Mapas por variables para horizonte a 7 días

Antes de iniciar los scripts que generan los mapas, debe verificar que se encuentren instaladas las librerías de R: raster, rgeos, rgdal y shape. La instalación la debe realizar en R desde una terminal e indique:

```
>install.packages("raster")
```

Cuando finalice, salga de R con: `q()` e indique n. Esto lo deberá hacer para las 3 instalaciones restantes.

Es posible que la instalación de rgdal genere conflicto en el equipo, por lo cual, en la carpeta de instaladores, se ubica el instalador. Para ejecutarlo desde la terminal deberá indicar:

```
>install.packages("home/ideam/Filtrokalman/instaladores/rgdal_1.2-20.tar.gz")
```

Cuando finalice, salga de R con: `q()` e indique n.

Una vez instalada la librería proceda con:

2.7.1 generarMapasKalmanPrec-v1.R

Este es el script que contiene toda la funcionalidad de los mapas para la precipitación. Inicialmente, identifica la fecha actual para ejecutar el algoritmo. En las filas 6-9 se ubican las rutas de fecha, año, mes, día y horagfs. Adicionalmente en la fila 10, se indica la ubicación de la carpeta de datos.

```
#6 anyo=commandArgs(TRUE)[1]
#7 mes=commandArgs(TRUE)[2]
#8 dia=commandArgs(TRUE)[3]
#9 horagfs=commandArgs(TRUE)[4]
#10 carpetadatos=commandArgs(TRUE)[5]
```

Para correr los mapas en una fecha específica, deberá inhabilitar las filas 6-10 y habilitar las del 12-16, indicando el año, mes (para meses menores a 10, colocar entre "" el dato) y día.

```
#12 anyo=2019
#13 mes="09"
#14 dia=18
#15 horagfs=12
#16 carpetadatos="/home/ideam/Filtro_kalman"
```

En las fila 19-21, se lee el archivo de coordenadas de puntos.

```
#19estaciones0=read.table(paste(carpetadatos,"datos_en
tradas/puntos.txt", sep = ""),header=FALSE, dec = ",",
sep="\t")
#21 nestacs=nrow(estaciones0)
```

En la fila 24 se ubica la ruta donde se ubicarán los mapas

```
#24 carpetasalidas=paste(carpetadatos,"salidasKalman",sep="")
```

Para generar los mapas, se debe contar con una tabla donde se ubiquen los valores a proyectar, para este

caso los datos diarios de los horizontes 1 hasta el 7. Este proceso se realiza desde la fila 28 hasta la 32.

```
#28 datosmapa=matrix(NA,nrow=nestacs,ncol=10)
#29 colnames(datosmapa)=c("Longitud","Latitud","
Estacion","Dia1","Dia2","Dia3","Dia4","Dia5","Di
a6","Dia7")
#30 datosmapa[,1]=estaciones0[,1]
#31 datosmapa[,2]=estaciones0[,2]
#32 datosmapa[,3]=estaciones0[,4]
```

En las filas 35 a la 42, se traen los datos desde las series generadas por el Filtro de Kalman.

El procesamiento de los mapas se inicia desde la fila 47. Inicialmente se carga el shp límite, para este caso se empleará el shp de los departamentos de Colombia y se deberá indicar su ubicación. Para este trabajo, se creó en datos_entradas, una subcarpeta llamada shp y dentro de ella se ubica el shp de departamentos. Es importante que el usuario cambie la ruta en la fila 47.

```
#47 mapa0<-readOGR("/home/jennifer/Filtro_kalman/datos_entradas/shp
/GBDIDEAM_departamentos", "GDBIDEAM_departamentos")
#48 mapa <- sTransform(mapa0, CRS("+proj=longlat"))
```

Las rutas de salida se indican en las filas 51-53

```
#51 archivoMapa=paste("Prec/mapas/GFSKalman_",anyo,mes,dia,"_",hor
agfs,"Z.jpg",sep="")
#52 archgraf=paste(carpetasalidas,"/",archivoMapa,sep="")
#53 jpeg(archgraf,width=1500,height=1000,units="px",quality=100,
bg="white")
```

Buscando una visualización ligera de los resultados, se realizó una ventana gráfica de 8 espacios, correspondientes a 4 columnas 2 filas, 7 para los mapas de los 7 horizontes y otra para la leyenda. Entre las filas 57 y 59 se realiza el procedimiento.

```
#57 par(mfrow=c(2,4),oma=c(0,0,5,0))
```

```
#59 for(i in 1:7){ # Para cada día pronosticado for(i
in 1:1){
```

Posteriormente, se grafica en la fila 63, el límite del mapa, para este caso los departamentos.

```
#63 plot(mapa,xlim=c(-85,-65),ylim=c(-5,15),
main=paste("Día ",i,sep=""),cex.main=3)
```

Desde la fila 66 hasta la 99, se deberá ajustar el color respectivo al rango. Tenga presente que el color debe ser indicado con código hexagecimal².

En la fila 102, se ubica el punto del Aeropuerto con coordenada y color respectivo al valor pronosticado según los rangos indicados.

```
#102 points(datosmapa[j,1], datosmapa[j,2],
col=colorpunto,lwd=1,pch=19,cex=2)
```

La leyenda se ubica en la caja octava en la zona del gráfico. Para ello, se genera un plot vacío. En la fila 109 realiza esta acción.

```
#109 plot(mapa,xlim=c(-85,-65),ylim=c(-5,15),
col="white",border="white")
```

Entre las filas 112 a la 121, se indicarán los rangos de la leyenda.

```
#112 legend("left",rev(c("0 a 0.5 mm","0.5 a 1 mm","1 a 3 mm","3 a 5
mm","5 a 10 mm",
#113 "10 a 15 mm","15 a 20 mm","20 a 25 mm","25 a 30 mm","30 a 35
mm",
#114 "35 a 40 mm","40 a 50 mm","50 a 75 mm","75 a 100 mm","más de
100 mm")),
#115 col=rev(c("#FA3C3C","#FF4600","#FF9600","#FFC800","#FFFA0
0","#00F000",
#116 "#00D200","#00A000","#007800","#DCDCFF","#BEBEFF",
"#9696FF",
#117 "#6464F0","#4646C8","#282896","#000000")),
#118 bty='n',cex=2.5,pch=19,pt.cex=3,y.intersp=1,
#119 title="Leyenda",title.adj=0.5)
```

El título del mapa se indica en la fila 121

```
#121 mtext(paste("Precipitación - ",anyo,mes,dia," -
GFS-Kalman",sep=""),outer=TRUE,cex=3.5)
Finalmente se exportan los mapas en archivo jpg en
la fila 124.
```

```
#124 dev.off()
```

2.7.2 generarMapasKalmanTemps-v1.R

Este es el script que contiene toda la funcionalidad de los mapas para las temperaturas mínimas, medias y máximas. Inicialmente, identifica la fecha actual para ejecutar el algoritmo. En las filas 6-9 se ubican las rutas de fecha, año, mes, día y horagfs. Adicionalmente en la fila 10, se indica la ubicación de la carpeta de datos.

```
#6 anyo=commandArgs(TRUE)[1]
#7 mes=commandArgs(TRUE)[2]
#8 dia=commandArgs(TRUE)[3]
#9 horagfs=commandArgs(TRUE)[4]
#10 carpetadatos=commandArgs(TRUE)[5]
```

² Puede consultar <https://www.colorhexa.com/>

Para correr los mapas en una fecha específica, deberá inhabilitar las filas 6-10 y habilitar las del 12-16, indicando el año, mes (para meses menores a 10, colocar entre "" el dato) y día.

```
#12 anyo=2019
#13 mes="09"
#14 dia="18"
#15 horagfs=12
#16 carpetadatos="/home/ideam/Filtro_kalman"
```

En las fila 19-21, se lee el archivo de coordenadas de puntos.

```
#19estaciones0=read.table(paste(carpetadatos, "/datos_en
tradas/puntos.txt", sep = ""),header=FALSE, dec = ",",
sep="\t")
#21 nestacs=nrow(estaciones0)
```

En la fila 24 se ubica la ruta donde se ubicarán los mapas

```
#24 carpetasalidas=paste(carpetadatos, "/salidasKalman", sep="")
```

Para generar los mapas, se debe contar con una tabla donde se ubiquen los valores a proyectar, para este caso los datos diarios de los horizontes 1 hasta el 7. Este proceso se realiza desde la fila 28 hasta la 32.

```
#28 varmet=c("TMed", "TMax", "TMin")
#29varmet2=c("Temperatura Media", "Temperatura
Máxima", "Temperatura Mínima")
```

```
#31for(z in 1:3){ # Para cada temperatura
En las filas 35-39, se crea la tabla con los datos
proyectados para los horizontes 1 al 7.
```

En las filas 42 a la 49, se traen los datos desde las series generadas por el Filtro de Kalman.

El procesamiento de los mapas se inicia desde la fila 54. Inicialmente se carga el shp límite, para este caso se empleará el shp de los departamentos de Colombia y se deberá indicar su ubicación. Para este trabajo, se creó en datos_entradas, una subcarpeta llamada shp y dentro de ella se ubica el shp de departamentos. Es importante que el usuario cambie la ruta en la fila 54.

```
#54mapa0<-readOGR("/home/jennifer/Filtro_kalman/datos_entradas/shp
/GBDIDEAM_departamentos", "GDBIDEAM_departamentos")
#55mapa <- spTransform(mapa0, CRS("+proj=longlat"))
```

Las rutas de salida se indican en las filas 58-60

```
#58archivoMapa=paste(varmet[z], "/mapas/GFSKalman_", anyo, mes, dia, "_
", horagfs, "Z.jpg", sep="")
#59archgraf=paste(carpetasalidas, "/", archivoMapa, sep = "")
```

```
#60jpeg(archgraf, width = 1500, height = 1000, units = "px", quality = 100,
bg = "white")
```

Buscando una visualización ligera de los resultados, se realizó una ventana gráfica de 8 espacios, correspondientes a 4 columnas 2 filas, 7 para los mapas de los 7 horizontes y otra para la leyenda. Entre las filas 64-66 se realiza el procedimiento.

```
#64 par(mfrow=c(2,4), oma=c(0, 0, 5, 0))
```

```
#66 for(i in 1:7){ # Para cada día pronosticado for(i
in 1:1){
```

Posteriormente, se grafica en la fila 70, el límite del mapa, para este caso los departamentos.

```
#70 plot(mapa,xlim=c(-85,-65),ylim=c(-5,15),
main=paste("Día ", i, sep=""), cex.main=3)
```

Desde la fila 73 hasta la 128, se deberá ajustar el color respectivo al rango. Tenga presente que el color debe ser indicado con código hexagecimal³.

En la fila 131, se ubica el punto del Aeropuerto con coordenada y color respectivo al valor pronosticado según los rangos indicados.

```
#131 points(datosmapa[j,1], datosmapa[j,2],
col=colorpunto, lwd=1, pch=19, cex = 2)
```

La leyenda se ubica en la caja octava en la zona del gráfico. Para ello, se genera un plot vacío. En la fila 138 realiza esta acción.

```
#138 plot(mapa,xlim=c(-85,-65),ylim=c(-5,15),
col="white", border="white")
```

Entre las filas 141 a la 149, se indicarán los rangos de la leyenda.

```
#112 legend("left",rev(c("0 a 0.5 mm", "0.5 a 1 mm", "1 a 3 mm", "3 a 5
mm", "5 a 10 mm",
#113 "10 a 15 mm", "15 a 20 mm", "20 a 25 mm", "25 a 30 mm", "30 a 35
mm",
#114 "35 a 40 mm", "40 a 50 mm", "50 a 75 mm", "75 a 100 mm", "más de
100 mm")),
#115col=rev(c("#FA3C3C", "#FF4600", "#FF9600", "#FFC800", "#FFFA0
0", "#00F000",
#116 "#00D200", "#00A000", "#007800", "#DCDCFF", "#BEBEFF",
"#9696FF",
#117 "#6464F0", "#4646C8", "#282896", "#000000")),
#118 bty='n', cex = 2.5, pch=19, pt.cex=3, y.intersp=1,
#119 title="Leyenda", title.adj = 0.5)
```

```
#141legend("left",rev(c("menor a 1.5°C", "1.5 a 3°C", "3 a 4.5°C", "4.5 a
6°C", "6 a 7.5°C", "7.5 a 9°C", "9 a 10.5°C",
#142 "10.5 a 12°C", "12 a 13.5°C", "13.5 a 15°C", "15 a 16.5°C", "16.5 a
18°C", "18 a 19.5°C",
#143 "19.5 a 21°C", "21 a 22.5°C", "22.5 a 24°C", "24 a 25.5°C", "25.5 a
27°C", "27 a 28.5°C",
```

³ Puede consultar <https://www.colorhexa.com/>

```

#144 "28.5 a 30°C","30 a 31.5°C","31.5 a 33°C","33 a 34.5°C","34.5 a
36°C","36 a 37.5°C",
#145 "37.5 a 39°C","más de 39°C"),
#146 col=rev(c("#FFFFFF","#E1FFF","#B4F0FA","#96D2FA",
"#78B9FA","#50A5F5","#3C96F5","#2882F0","#1E6EEB",
#147 "#1464D2","#E6FFE1","#C8FFBE","#B4FAAA","#96F58C",
"#78F573","#50F050","#37D23C","#1EB41E",
#148 "#0FA00F","#FFFAAA","#FFE878","#FFC03C","#FFA000",
"#FF6000","#FF3200","#E11400","#C00000")),
#149 bty='n', cex = 2.2, pch=19, pt.cex=3, y.intersp=0.75, ncol=2,
title="Leyenda", title.adj = 0.5)

```

```

El título del mapa se indica en la fila 151
#151 mtext(paste("Precipitación - ",anyo,mes,dia," - GFS-
Kalman",sep=""), outer = TRUE, cex = 3.5)
Finalmente se exportan los mapas en archivo jpg en la fila
154.
#154 dev.off()

```

3 RESULTADOS

La aplicabilidad del Filtro de Kalman sobre las variables precipitación, temperatura media, temperatura mínima y temperatura máxima generó un resultado confiable y satisfactorio para las series. En primer lugar, se presentará un comparativo en los resultados obtenidos para las temperaturas (tabla 1) en el mes de agosto para la variable temperatura mínima, generadas a inicio de cada semana (5, 12, 19 y 26) con horizonte a 7 días, para los Aeropuertos de El Dorado y El Carano.

Tabla 1. Temperatura mínima.

Estación 11045010 - Apto El Carano							
	Horz 1	Horz 2	Horz 3	Horz 4	Horz 5	Horz 6	Horz 7
Temperatura mínima	Día 5	6	7	8	9	10	11
	Proy. GFS	22	22	21	23	22	22
	GFS-Kalman	24,8	23,4	22,6	22,9	23,6	23,9
	Observado	24,6	23	22,4	22,8	23,4	23,2
	Día 12	13	14	15	16	17	18
	Proy. GFS	22	22	22	21	19	19
	GFS-Kalman	24,2	24,1	24,6	23,6	21,7	21,7
	Observado	24	23,8	24,4	23,2	21,4	21,6
	Día 19	20	21	22	23	24	25
	Proy. GFS	22	22	21	22	22	22
	GFS-Kalman	23,6	23,8	23,7	22,9	23,8	24,3
	Observado	23,4	23,4	23,4	22,6	23,7	24
Día 26	27	28	29	30	31	1	
Proy. GFS	22	22	22	22	22	22	
GFS-Kalman	24,2	25,3	24,1	24,2	24,9	24,9	
Observado	24	25,1	23,6	24,1	23,4	23,6	

Estación 2120579 - Apto El Dorado							
	Horz 1	Horz 2	Horz 3	Horz 4	Horz 5	Horz 6	Horz 7
Temperatura mínima	Día 5	6	7	8	9	10	11
	Proy. GFS	6	6	7	7	6	7
	GFS-Kalman	11,3	12,5	9,8	10	9,9	8,1
	Observado	11,2	12,1	9,3	10	9,6	7,7
	Día 12	13	14	15	16	17	18
	Proy. GFS	7	9	8	7	8	6
	GFS-Kalman	9,9	10,7	11,7	10,3	8,3	9
	Observado	9,6	10,5	11,4	9,8	8	10,3
	Día 19	20	21	22	23	24	25
	Proy. GFS	6	7	5	6	8	6
	GFS-Kalman	10,8	7,3	9,8	5,9	9,4	11,7
	Observado	10,4	6,8	9,9	5,2	9,7	11,3
Día 26	27	28	29	30	31	1	
Proy. GFS	6	6	6	6	6	9	

	GFS-Kalman	6,9	7,5	10,5	9,4	9,7	14,1	12,6
Observado		6,9	7,4	10,4	8,8	4,6	7,2	9,8

Como se pudo observar, la temperatura mínima GFS-Kalman, presentó buen ajuste tanto para el horizonte 1 hasta el 7. A continuación, en la figura 2, se presentan las salidas gráficas que genera el Filtro para estos puntos analizados.

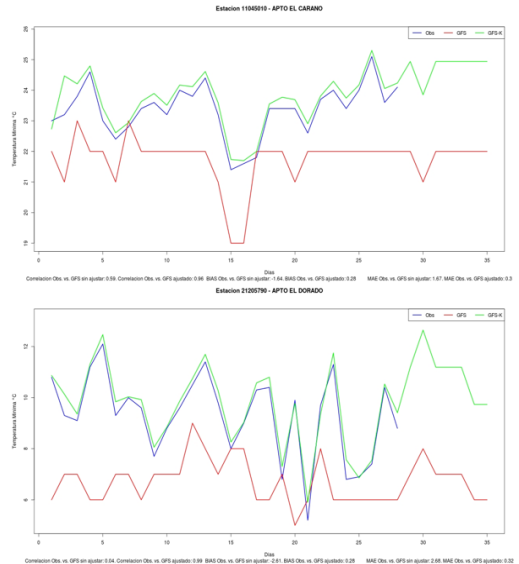


Figura 2. Distribución espacial de los Aeropuertos evaluados con salidas del modelo GFS

En la tabla 2, se presentan los resultados de las salidas del mes de agosto para la variable temperatura media, generadas a inicio de cada semana (5, 12, 19 y 26) con horizonte a 7 días, para los Aeropuertos de El Dorado y El Carano.

Tabla 2. Temperatura media.

Estación 11045010 - Apto El Carano							
	Horiz 1	Horiz 2	Horiz 3	Horiz 4	Horiz 5	Horiz 6	Horiz 7
Temperatura media	Día 5	6	7	8	9	10	11
	Proy. GFS	27	25,6	26,6	27,9	27,5	28,2
	GFS-Kalman	24,5	23,5	24,8	24,2	23,8	24
	Observado	27,3	25,9	26,7	28,1	27,9	28,4
	Día 12	13	14	15	16	17	18
	Proy. GFS	26,2	26	26,1	24,2	23	26,6
	GFS-Kalman	24,2	23,2	23,8	23	21,2	21,2
	Observado	26,4	26,2	26,3	24,6	23,2	26,5
	Día 19	20	21	22	23	24	25
	Proy. GFS	24	27,7	26,5	28,3	28,2	26,2
	GFS-Kalman	24,2	24,5	24	24,8	24,8	24,2
	Observado	24,8	27,4	27	28,4	28,6	26,6
Día 26	27	28	29	30	31	1	
Proy. GFS	28,5	29,3	25,9	25,9	24,2	23,8	
GFS-Kalman	25,2	24	24	24	28,3	27,8	
Observado	28,7	29,6	26,5	25,9	27,7	26,1	

Observado	32,7	34,5	28,3	32	32,8	29	30,2
-----------	------	------	------	----	------	----	------

Estación 2120579 - Apto El Dorado

	Horiz 1	Horiz 2	Horiz 3	Horiz 4	Horiz 5	Horiz 6	Horiz 7
Día	5	6	7	8	9	10	11
Proy. GFS	9,5	10,2	10,5	10,8	10,5	10,5	11,2
GFS-Kalman	14	14,6	14,9	14,1	14,5	14,3	14,4
Observado	13,7	14,4	14,6	13,7	14,3	14	14,2
Día	12	13	14	15	16	17	18
Proy. GFS	11,5	10,8	10,8	9,2	11,2	11,2	10,8
GFS-Kalman	15,1	14,8	13,9	13,3	11,8	12,7	14,6
Observado	14,9	14,4	13,6	13	11,5	12,6	14,4
Día	19	20	21	22	23	24	25
Proy. GFS	10,2	10	10,2	11,5	11,2	10	10,8
GFS-Kalman	13,1	13,4	14,1	13,6	14,5	14,8	13,6
Observado	12,5	13,3	13,9	13,3	14,3	14,5	13,2
Día	26	27	28	29	30	31	1
Proy. GFS	10,8	10,8	10,8	10,8	11,5	12,5	10,8
GFS-Kalman	13,3	14,5	14,6	14,6	16,3	17,6	15,3
Observado	13,1	14,4	14,2	14,3	13,7	13,6	13,6

Como se pudo observar, la temperatura mínima GFS-Kalman, presentó buen ajuste tanto para el horizonte 1 hasta el 7. A continuación, en la figura 3, se presentan las salidas gráficas que genera el Filtro para estos puntos analizados.

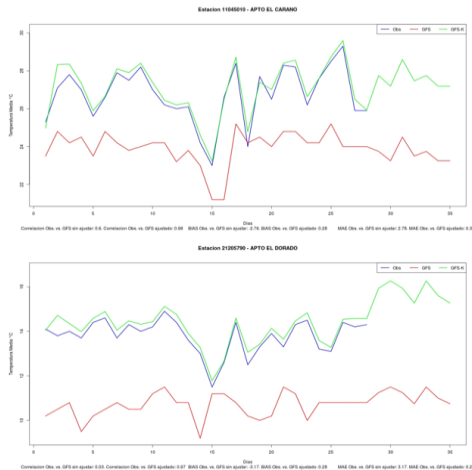


Figura 3. Distribución espacial de los Aeropuertos evaluados con salidas del modelo GFS

En la tabla 3, se presentan los resultados de las salidas del mes de agosto para la variable temperatura máxima, generadas a inicio de cada semana (5, 12, 19 y 26) con horizonte a 7 días, para los Aeropuertos de El Dorado y El Carano.

Tabla 3. Temperatura máxima.

Estación 1104510 - Apto El Carano

	Horiz 1	Horiz 2	Horiz 3	Horiz 4	Horiz 5	Horiz 6	Horiz 7
Día	5	6	7	8	9	10	11
Proy. GFS	30	26	30	31	30	31	28
GFS-Kalman	32,8	30,6	31,6	33,2	32,4	34	30
Observado	33	31,1	31,6	33,3	32,9	34,1	30,7
Día	12	13	14	15	16	17	18
Proy. GFS	29	29	31	26	23	23	33
GFS-Kalman	33,2	29,6	32,4	26,4	24,2	33	33
Observado	33	30,3	32,2	27,3	24,3	32,3	33,8
Día	19	20	21	22	23	24	25
Proy. GFS	30	31	30	32	33	31	32
GFS-Kalman	26	32,6	29,8	33,4	33,6	33,2	32,6
Observado	27	31,8	30,6	33,3	34	33,6	32,9
Día	26	27	28	29	30	31	1
Proy. GFS	32	28	28	28	33,5	31,3	32,4
GFS-Kalman	32,5	34,4	27,4	32,6	30	28	29

Estación 2120579 - Apto El Dorado

	Horz 1	Horz 2	Horz 3	Horz 4	Horz 5	Horz 6	Horz 7
Día	5	6	7	8	9	10	11
Proy. GFS	15	17	18	18	19	19	18
GFS-Kalman	18,5	19	19,8	20	19,7	20,2	20,1
Observado	18,1	18,9	19,6	19,7	19,4	20	19,8
Día	12	13	14	15	16	17	18
Proy. GFS	19	15	16	11	16	16	18
GFS-Kalman	20,5	20,1	18,2	17,1	16	16,2	20
Observado	20,3	19,7	17,8	16,9	15,8	16,1	20,1
Día	19	20	21	22	23	24	25
Proy. GFS	17	14	18	19	17	17	17
GFS-Kalman	17,1	19,6	19,4	20,5	19,3	18,9	19,3
Observado	16,3	19,8	19	20,3	18,8	18,7	19,1
Día	26	27	28	29	30	31	1
Proy. GFS	18	17	17	17	20	19	17
GFS-Kalman	19,4	20,1	18,7	19,9	20,7	19,6	19,1
Observado	19,1	19,9	18,3	19,9	20,7	18,1	18,9

Como se pudo observar, la temperatura máxima GFS-Kalman, presentó buen ajuste tanto para el horizonte 1 hasta el 7. A continuación, en la figura 4, se presentan las salidas gráficas que genera el Filtro para estos puntos analizados

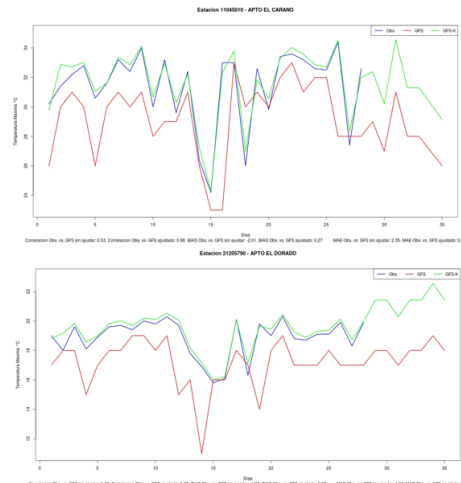


Figura 4. Distribución espacial de los Aeropuertos evaluados con salidas del modelo GFS

En cuanto a la variable precipitación, en noviembre se pudo observar que el Filtro de Kalman presentó un ajuste bastante bueno con el método Unscented. Sin embargo, también se encontró que con el método lineal ajusta muy bien pero por método basado en que la precipitación es una variable acumulativa y no lineal como la temperatura, se optó por realizar el ajuste. También es importante analizar que al aplicar este método, es importante tener días con anterioridad para buscar que el ajuste sea mejor en días próximos al horizonte 7.

		Estación 2120579 - Apto El Dorado						
		Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7
Precipitación	Día	11	12	13	14	15	16	17
	Observado	28,7	12,4	0	0,3	0	0	0
	Proy. GFS	14	7	2	4	12	10	11
	GFS-Kalman	28,2	14,7	0,1	0	0	0,9	1,1
	Día	18	19	20	21	22	23	24
	Observado	0	3,9	0	20,6	32,8	0	0
	Proy. GFS	5	6	11	12	19	14	10
	GFS-Kalman	1	4,6	0,7	19,6	33,5	3,9	0

		Estación 11045010 - Apto El Carano						
		Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7
Precipitación	Día	11	12	13	14	15	16	17
	Observado	9,1	36,7	7,9	53	33,2	0	48,8
	Proy. GFS	27	28	21	16	16	13	19
	GFS-Kalman	9,2	34,1	12,1	48,2	37,5	2,2	43
	Día	18	19	20	21	22	23	24
	Observado	3,8	0	8,5	51,8	0	30,5	9,2
	Proy. GFS	14	15	24	21	18	9	13
	GFS-Kalman	10,3	0	7,6	49,2	7,4	26,2	12,1

Otra de las salidas generadas en el Filtro, fueron los mapas. Para esta visualización, en la figura 5 se presentan las salidas de la precipitación para un horizonte a 7 días.

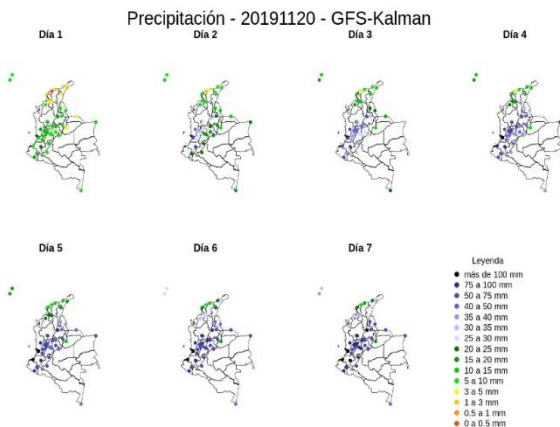


Figura 4. Distribución espacial de los resultados de pronóstico para la variable precipitación en los Aeropuertos evaluados con salidas del modelo GFS para un horizonte a 7 días

Aunque los resultados muestran una mejora significativa en los datos, es importante mencionar que esto se refleja casi en un 98% de los 42 puntos analizados. Adicionalmente, es importante anotar que el ajuste logra identificar cambios en las temperaturas abruptas de un día para otro, como es el caso de las temperaturas mínimas. Sin embargo, el error entre el proyectado y el observado alcanza a estar entre 1°C aproximadamente. Para este caso, se sugiere tener presente las caídas de valores de temperatura y analizar la probabilidad de un descenso de la mínima.

4 DISCUSIONES

Si bien el Filtro de Kalman es una muy buena herramienta de corrección de salidas de modelos como en este caso GFS, los mayores retos se basan en poder alinear la automatización de las entradas del modelo ya que dependen de que el servicio de internet y datos diarios de la intranet sean continuos.

En cuanto a la aplicabilidad del Filtro dentro del IDEAM, se indica que esta es una muy buena herramienta que mejora las salidas del GFS u otro modelo que cumpla con los parámetros de entrada, permitiendo que se tenga mejores datos en la predicción del estado del Tiempo.

5 CONCLUSIONES

El Filtro de Kalman aplicado en las temperaturas bajo un modelo de Filtro Lineal y para la precipitación Filtro de Kalman Unscented, permitió tener un buen resultado en todas las proyecciones. En ambos casos se logró cumplir con el objetivo de mejorar las salidas del modelo GFS. Sin embargo, dentro de las pruebas realizadas al inicio del proyecto, se encontró que el método de Linealidad es un poco más acertado en la variable precipitación para horizontes mayores a 5 días.

En cualquier aplicabilidad del método, se sugiere tener series históricas mínimo de 5 días histórico para un mejor ajuste en días proyectados mayores a 5 días, sin embargo, el Filtro ajusta solo con 1 dato de histórico.

6 AGRADECIMIENTOS

A los profesionales Franklyn Ruiz y Jeimmy Melo de la subdirección de Meteorología por el apoyo computacional y aporte científico y al Ing. Guillermo Armenta por su colaboración en el seguimiento en la elaboración del código.

7 BIBLIOGRAFÍA

Andrews, Grewal. Kalman Filtering Theory And Practice Using Matlab. New York: Wiley, 2001

Application of Numerical Weather Prediction. Objective weather interpretation with direct use of numerical model output. Japan International Cooperation Agency Japan Meteorological Agency. 2003.

Chacón, A. C. Evaluación de las salidas de los modelos meteorológicos GFS y MM5 para las zonas astronómicas de Paranal, Macón y Pachón. Universidad de Valparaiso de Chile. 2009.

Deossa, P.A. Entrenamiento de Redes Neuronales con Filtros de Kalman. Universidad Nacional de Colombia, Sede Medellín. 2009.

Dorado, J; Ruíz, J. F. Manual de usuario ApplyKalmanFilter_Linux. Grupo de Modelamiento de Tiempo y Clima. Subdirección de Meteorología. IDEAM 2013.

Haykin, Simon. Neural Networks. A comprehensive Foundation. Pearson, India, 1999.

Yamagishi, Y., 2003. Introduction to the numerical weather prediction with emphasis on the application to the short term weather forecasting. Japan Meteorological Agency. Octubre 2003. Tokyo, Japan.