



*Certificación ISO 9001:2008 ‡*

---

---

# **ANÁLISIS MEDIANTE UNA RED NEURONAL ARTIFICIAL DE LA RESPUESTA DEL PUERTO DE MANZANILLO BAJO EL EFECTO DE TSUNAMIS OCURRIDOS EN EL PASADO EN LA COSTA OCCIDENTAL DE MÉXICO**

José Miguel Montoya Rodríguez  
Juan Pedro Vásquez López

**Publicación Técnica No. 446  
Sanfandila, Qro. 2015**



---

**SECRETARÍA DE COMUNICACIONES Y TRANSPORTES**  
**INSTITUTO MEXICANO DEL TRANSPORTE**

**Análisis mediante una red neuronal artificial de la  
respuesta del Puerto de Manzanillo bajo el efecto  
de tsunamis ocurridos en el pasado, en la costa  
occidental de México**

**Publicación Técnica No.446**  
**Sanfandila, Qro. 2015**

---



Esta investigación fue realizada en la Coordinación de Ingeniería Portuaria Y Sistemas Geoespaciales del Instituto Mexicano del Transporte, por el Jefe de la División de Ingeniería de Puertos y Costas M. en C. José Miguel Montoya Rodríguez y el investigador titular M. en C. Juan Pedro Vásquez López.

# Contenido

---

Resumen		iii
Abstract		iv
Resumen ejecutivo		v
Capítulo 1.	Introducción	1
Capítulo 2.	Metodología	5
Capítulo 3.	Resultados	9
Capítulo 4.	Conclusiones	13
Bibliografía		15
Anexo	Rutinas de código	17

# Resumen

---

El presente estudio tuvo como objetivo diseñar, codificar y optimizar dos redes neuronales artificiales tipo feedforward, como estimadores de patrones de rapidez de corrientes en el interior del Puerto de Manzanillo en Colima, sujeto a la acción de tsunamis históricos, lográndose un aprendizaje efectivo a partir de tsunamis sintéticos locales, el patrón de rapidez máxima y el patrón de rapidez media, con un error porcentual absoluto medio del orden de 20% en ambos casos.



# Abstract

---

This study had the objective to design, coding and optimizing two feedforward-type artificial neural networks, as estimators of current speed patterns inside the Port of Manzanillo in Colima, with the action of historical tsunamis, achieving effective learning from local synthetic tsunamis, the maximum speed pattern and the average speed pattern, with a mean absolute percentage error of the order of 20% in both cases.



# Resumen ejecutivo

---

Las redes neuronales artificiales (RNA) son algoritmos computacionales que han ganado popularidad resolviendo problemas que involucran muchas variables, de naturaleza no lineal o cuando no existía un modelo previo que describiera las relaciones funcionales entre dichas variables.

Estas RNA son sistemas de procesamiento de datos e información modelados en similitud del funcionamiento de un cerebro biológico real. Tienen la capacidad de procesar en paralelo grandes cantidades de información, aun cuando ésta sea parcial y difusa. Pueden aprender y memorizar información muy variada y formalizarla, permitiendo hacer predicciones a partir de los datos con los que ha sido entrenada.

Se reconoce que las RNA se han aplicado exitosamente en diversos problemas en el campo de la ingeniería marítima y de costas. Sin embargo, al momento de realizar este trabajo, son pocas las fuentes bibliográficas que abordan el tema de RNA aplicadas a tsunamis. Menos aún a tsunamis ocurridos en la costa occidental de México.

El presente trabajo consideró 50 puntos geográficos de modelación sobre la entrada del Puerto de Manzanillo. Junto con seis casos de tsunamis locales sintéticos, a partir de los cuales se entrenaron un par de redes neuronales artificiales tipo feedforward, una para el patrón de rapidez máxima de corriente y otra para el patrón de rapidez media, respectivamente. Teniendo como objetivo de aprendizaje, los principales tsunamis ocurridos en el pasado en dicho puerto.

La mayor parte del código está desarrollado en la plataforma estadística R., haciendo uso de técnicas de cómputo paralelo para acortar el tiempo de cálculo de las miles de corridas involucradas y almacenando todos los datos en una base de datos relacional.

Se obtuvieron así un par de redes neuronales artificiales que aprendieron efectivamente los patrones antes mencionados con una precisión del orden del 20% de error porcentual absoluto medio.



# 1 Introducción

---

Las redes neuronales artificiales (RNA) son algoritmos computacionales que han ganado popularidad resolviendo problemas que involucran muchas variables, de naturaleza no lineal o cuando no existía un modelo previo que describiera las relaciones funcionales entre dichas variables [1, 2].

Estas RNA son sistemas de procesamiento de datos e información modelados en similitud del funcionamiento de un cerebro biológico real. Tienen la capacidad de procesar en paralelo grandes cantidades de información, aun cuando ésta sea parcial y difusa. Pueden aprender y memorizar información muy variada y formalizarla, permitiendo hacer predicciones a partir de los datos con los que ha sido entrenada.

En el campo de la ingeniería marítima y de costas, se reconoce que las RNA se han aplicado exitosamente en problemas que involucran la estimación o predicción de variables ambientales, comportamiento de estructuras, predicción de niveles de marea y oleaje, predicción de tormentas, morfología de costas, entre otras; llevando a cabo tareas de aproximación de funciones, optimización, modelado de sistemas y aprendizaje de patrones [3].

Sin embargo, al momento de realizar este trabajo, son pocas las fuentes bibliográficas que abordan el tema de RNA aplicadas a tsunamis. Menos aún a tsunamis ocurridos en la costa occidental de México.

En México, los estados de Colima, Guerrero, Oaxaca y Michoacán son los que han presentado mayor incidencia de tsunamis locales a lo largo de la historia reciente. El recuento documental de afectaciones por tsunamis en México comprende casos a lo largo de la Costa Occidental desde el año 1787 al 2003. Teniéndose verificados 60 tsunamis en 250 años [5], distinguiéndose dos tipos: los de origen lejano, como por ejemplo los recientes tsunamis originados en Sumatra en el año 2004, Chile 2010 y Japón 2011; cuyas consecuencias incluyen alturas de ola de un máximo de 2.5 m y, por otra parte, los tsunamis de origen cercano, que han llegado a tener alturas de ola de 5 y 10 m de altura; como los que se registraron en las costas de Oaxaca en 1787 y en el estado de Guerrero en 1925.

Los tsunamis ocurridos en México han tenido una mediana de 2 m de altura de ola, pero algunos casos extremos llegan a superar la barrera de los 10 m y, por lo general, son generados por sismos grandes, mayores de 7.0, en particular, de mediana 7.7 grados en la escala sísmica de Richter.

Sin embargo, hay el antecedente de una investigación referente al tema y que servirá de base fundamental al presente trabajo. Llevada a cabo por personal de la División de Ingeniería de Puertos y Costas del IMT, detalla el diseño, programación y optimización de un par de redes neuronales artificiales tipo *feedforward* (RNAf) que reproducen los patrones de rapidez máxima y media de las corrientes respectivamente, que se dan en el interior del Puerto de Manzanillo bajo la acción de tsunamis (Proyecto VI-17/13) [6].

En dicho trabajo se consideraron 50 puntos geográficos de modelación sobre la entrada del Puerto de Manzanillo, como muestra la Figura 1.1. Además de 21 casos de elevación inicial del tsunami sobre la línea batimétrica 34 metros del puerto, como muestra la Tabla 1.1.

Es importante mencionar que los datos de velocidades de corrientes, proceden a su vez de un estudio anterior (Proyecto VE-12/13) [7] donde se empleó un modelo hidráulico a escala 1:150, mostrado en la Figura 1.2.

**Tabla 1.1 Casos de elevación inicial en la línea batimétrica 34m del Puerto de Manzanillo, Col. (Proyecto de investigación VI-17/13)**

Caso Núm.	1	2	3	4	5	6	7	8	9
$\eta_0$ (m)	1.07	1.08	1.24	1.42	1.46	1.60	1.78	2.49	2.58

Caso Núm.	10	11	12	13	14	15	16	17	18
$\eta_0$ (m)	2.66	2.84	3.02	3.20	3.37	3.44	3.55	4.30	6.02

Caso Núm.	19	20	21
$\eta_0$ (m)	8.60	12.05	15.50



**Figura 1.1** Puntos de modelación del Puerto de Manzanillo para rapidez máxima y media de corrientes en el *Proyecto de investigación VI-17/13*



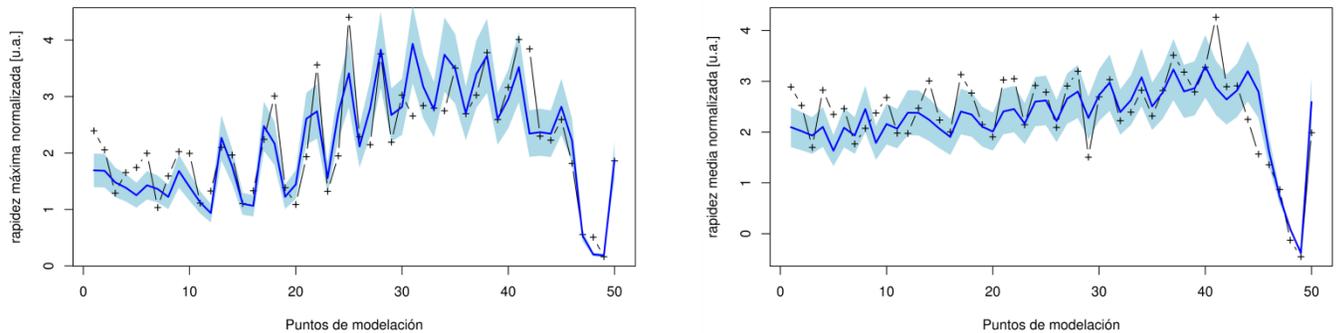
**Figura 1.2** Modelo hidráulico a escala del Puerto de Manzanillo, utilizado en el *Proyecto de investigación VE-12/13* que aportó datos para el *Proyecto de investigación VI-17/13*

Las RNAf optimizadas se entrenaron con los primeros 20 casos de elevación inicial. Teniendo como variables de entrada:

- el número del punto de modelación
- la elevación inicial
- la latitud, la longitud y la profundidad del punto de modelación

Con una sola variable de salida: la rapidez máxima o media, según el caso.

Mostraron un aprendizaje efectivo del patrón de corrientes, obteniendo menos de 20% de error al contrastar las predicciones hechas con la RNAf contra el último caso de elevación inicial. Resultados satisfactorios considerando la baja cantidad total de casos de entrenamiento disponibles.



**Figura 1.3 Comparación de patrones de rapidez máxima (izquierda) y rapidez media (derecha) de corrientes en el interior del Puerto de Manzanillo del Proyecto de investigación VI-17/13. Los puntos señalados con ‘+’ corresponden a los datos del modelo físico, la línea azul continua es la predicción de la RNAf y la sombra azul claro es el error porcentual absoluto medio entre ambos**

## 2 Metodología

---

Ante tal panorama, como una necesaria continuación de los trabajos en este tema, el presente estudio tiene por objetivos desarrollar una red neuronal tipo *feedforward*; y explorar la posibilidad de aplicar ésta al análisis de la respuesta del Puerto de Manzanillo, Colima, bajo el efecto de tsunamis ocurridos en el pasado en la Costa Occidental de México.

Se utilizaron los datos sísmicos y de velocidad de corrientes aportados por el Proyecto de Investigación VE-12/13 [7], correspondientes a los tres principales tsunamis históricos en el Puerto de Manzanillo, Colima, como se listan en la Tabla 2.1.

**Tabla 2.1 Tsunamis históricos y la elevación inicial en la línea batimétrica 34m del Puerto de Manzanillo, Col. (Proyecto VE-12/13)**

Caso Núm.	1	2	3
Fecha del sismo	1985-09-19	1932-06-03	1995-10-19
$\eta_0$ (m)	1.08	1.46	3.44
Momento sísmico (Mw)	8.1	8.2	8.1
Dislocación (m)	3.3	2.4	7

Con el fin de buscar un par de RNaf que aprendan efectivamente los patrones de rapidez máxima y media de corrientes generadas en el interior del puerto durante dichos tsunamis históricos, se consideraron como casos de entrenamiento para las redes, seis tsunamis sintéticos locales aportados por el Proyecto de Investigación VI-17/13 [6], listados en la Tabla 2.2.

**Tabla 2.2 Tsunamis sintéticos locales y la elevación inicial en la línea batimétrica 34m del Puerto de Manzanillo, Col. (Proyecto VI-17/13)**

Caso Núm.	1	2	3	4	5	6
$\eta_0$ (m)	2.58	4.30	6.02	8.60	12.05	15.50

Para hallar las RNaf óptimas, se definieron los siguientes conjuntos dominio, resumidos en la Tabla 2.3, que incluyen las posibles variantes de los parámetros de las redes, siendo la mejor combinación la que resulte en un menor error porcentual

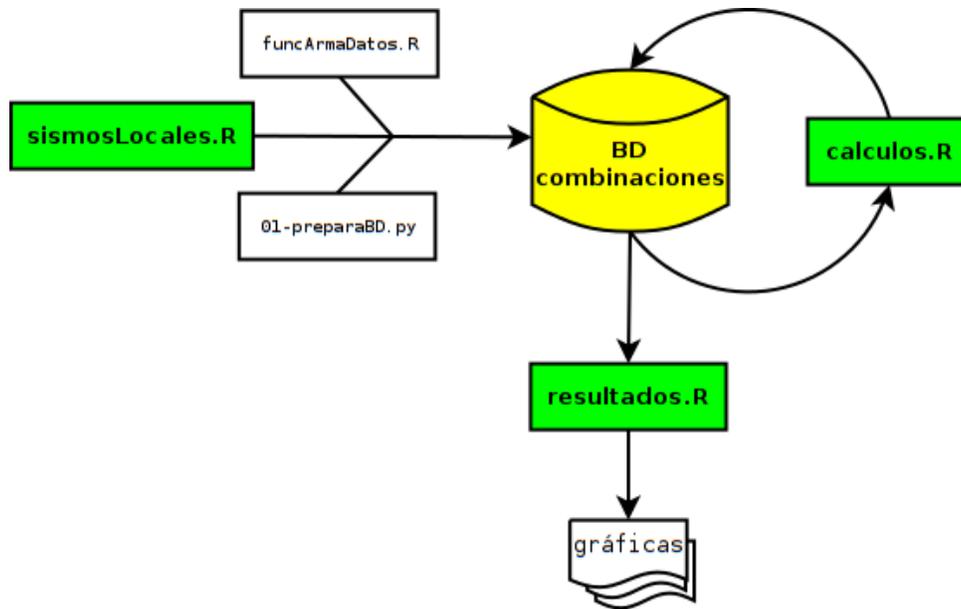
absoluto medio (MAPE) entre el patrón de rapidez del tsunami histórico y la respuesta de la red.

**Tabla 2.3 Conjuntos dominio de parámetros de las RNAf**

Variables de entrada	Variables de salida	Tasa de aprendizaje	Momentum global	Ciclos de entrenamiento	Factor para el número de neuronas en capa oculta
	0. Rapidez	0.02	0.05	$1 \times 10^3$	2
	0. Rapidez 1. Número de punto	0.002	0.005	$9 \times 10^3$	2.25
1. Número de punto 2. Elevación inicial 3. Longitud 4. Latitud 5. Profundidad	0. Rapidez 1. Número de punto 2. Elevación inicial	$2 \times 10^{-4}$	$5 \times 10^{-4}$	$9 \times 10^4$	2.5
	0. Rapidez 1. Número de punto 5. Profundidad	$2 \times 10^{-5}$	$5 \times 10^{-5}$		2.75
	0. Rapidez 2. Elevación inicial 5. Profundidad				3
1. Número de punto 2. Elevación inicial	0. Rapidez 1. Número de punto 2. Elevación inicial 5. Profundidad				3.25
1. Número de punto 2. Elevación inicial 5. Profundidad					

De donde claramente se reconocen del orden de  $5 \times 10^3$  corridas para cada uno de los tipos de rapidez de corriente. Por lo que, para acortar los tiempos de cálculo, es necesario implementar estrategias de cómputo paralelo, y así aprovechar toda la capacidad de procesamiento de los varios núcleos con los que cuentan los CPUs actuales, como con el que cuenta la computadora donde se llevaron a cabo las corridas.

El código está desarrollado principalmente en 3 rutinas R: la primera almacena todas las combinaciones de los conjuntos dominio de parámetros de las RNAf en una base de datos relacional construida en MySQL<sup>®</sup>, la segunda, hace propiamente el entrenamiento y simulación de la RNAf, almacenando sus resultados en la misma base de datos, y una última, grafica los resultados, como se describe en la Figura 2.1.



**Figura 2.1** Flujo de trabajo de las rutinas que buscan las RNAf óptimas para los patrones de rapidez máxima y media de corrientes



### 3 Resultados

Las RNAf óptimas que aprendieron los patrones de rapidez máxima y media de corrientes al interior del Puerto de Manzanillo, sobre los 50 puntos de modelación mostrados en la Figura 1.1, bajo la acción de los tsunamis históricos de la Tabla 2.1, tienen las siguientes características.

#### 3.1 RNAf estimador del patrón de rapidez máxima

Para el patrón de rapidez máxima, el aprendizaje fue efectivo, con un error de apenas 17.65%, como se muestra en las gráficas de linealidad y normalidad de la Figura 3.1.

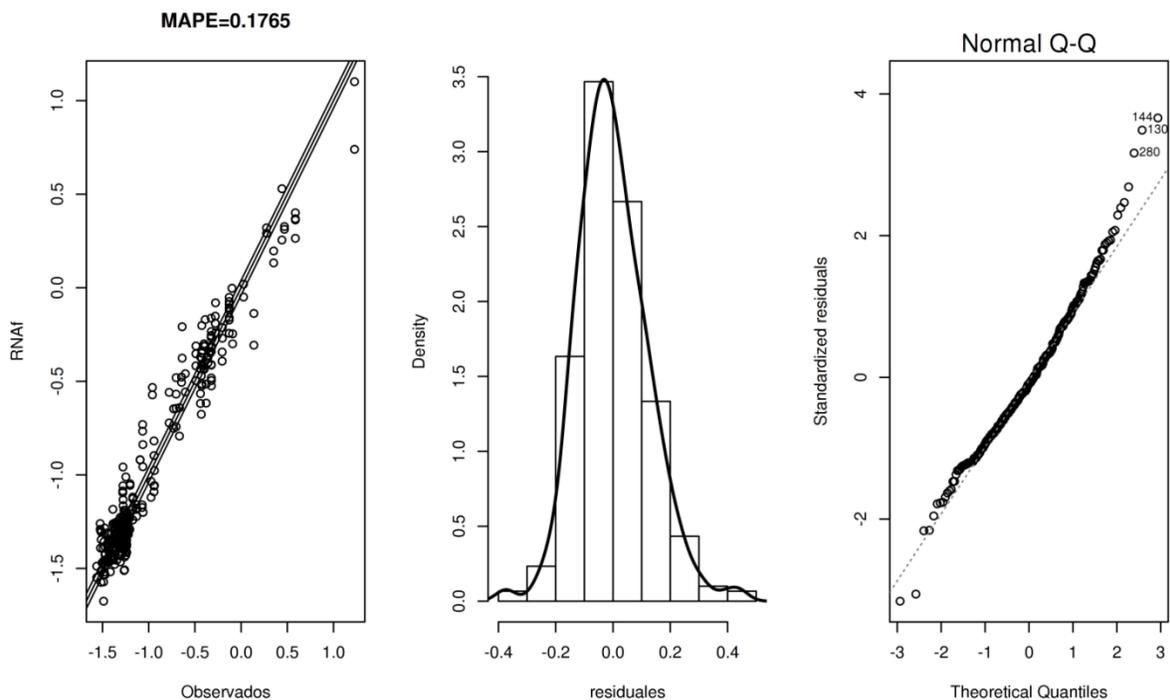


Figura 3.1 Gráficas de linealidad y normalidad de la respuesta de la RNAf del patrón de rapidez máxima

Su topología es de 5 variables de entrada: 1) número de punto de modelación, 2) elevación inicial ( $\eta_0$ ), 3) latitud, 4) longitud y 5) profundidad z del punto de modelación; y dos variables de salida: 1) la rapidez máxima y 2) el número de punto de modelación, con 17 neuronas en la capa intermedia y  $9 \times 10^4$  ciclos de entrenamiento. Como muestra el diagrama de la Figura 3.2.

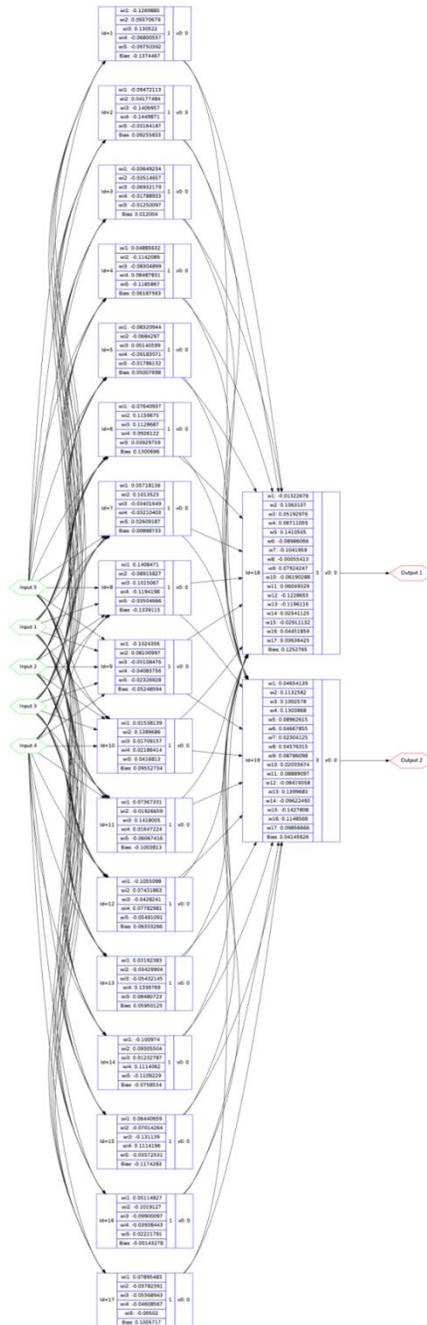
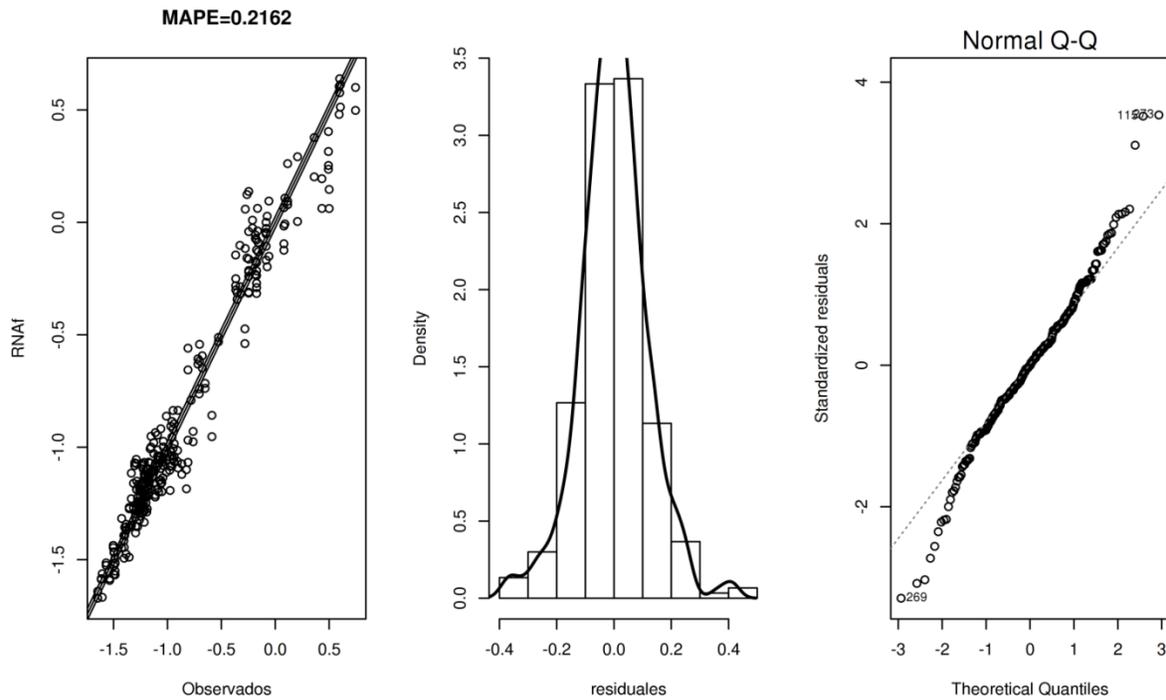


Figura 3.2 Topología de la RNAf del patrón de la rapidez máxima

## 3.2 RNAf estimador del patrón de rapidez media

En el caso de rapidez media, el menor MAPE alcanzado fue de 21.62%, como se muestra en la Figura 3.3.



**Figura 3.3 Gráficas de linealidad y normalidad de la respuesta de la RNAf del patrón de rapidez media**

En este caso, la topología es de 5 variables de entrada: 1) número de punto de modelación, 2) elevación inicial ( $\eta_0$ ), 3) latitud, 4) longitud y 5) profundidad z del punto de modelación; y una variables de salida: 1) la rapidez media, con 17 neuronas en la capa intermedia y  $9 \times 10^4$  ciclos de entrenamiento, como muestra el diagrama de la Figura 3.4.

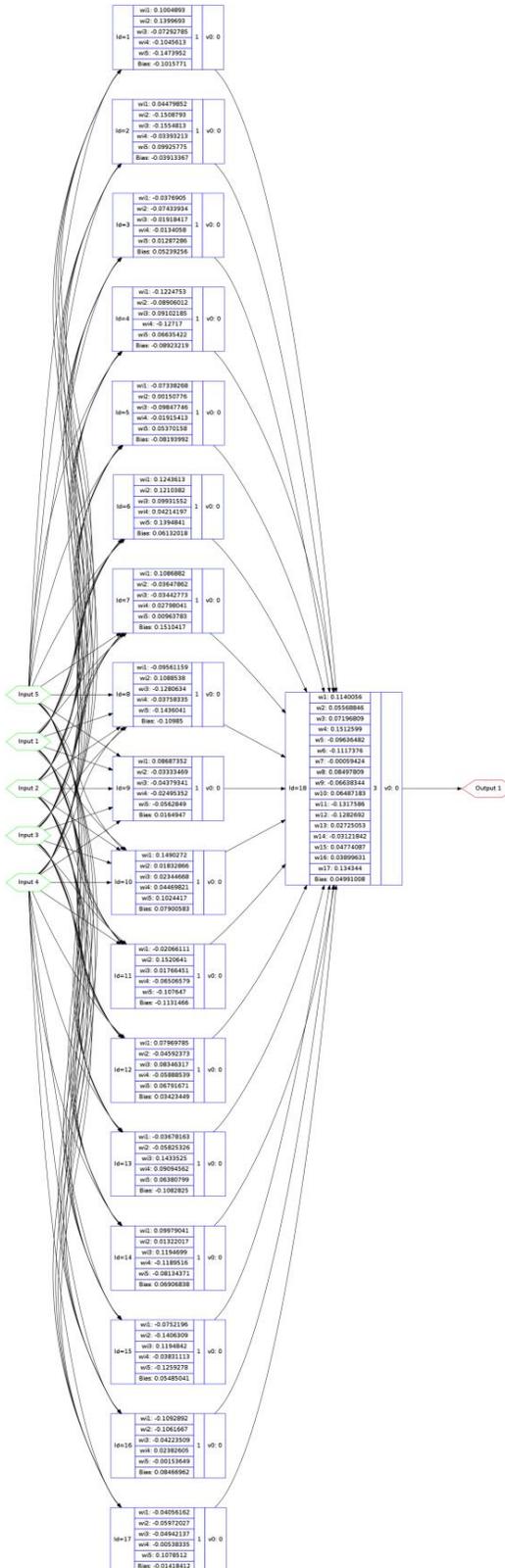
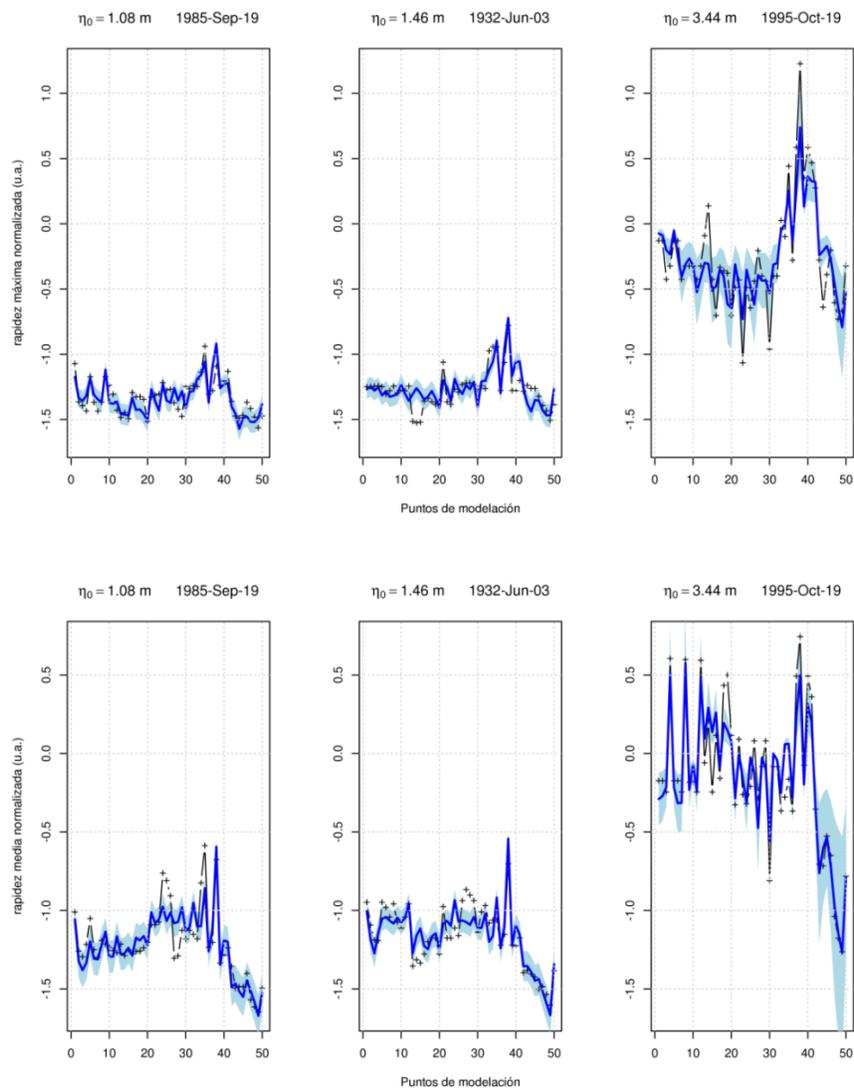


Figura 3.4 Topología de la RNAf del patrón de la rapidez media

## 4 Conclusiones

Se logró entonces, construir un par de RNAf, que aprendieron los patrones de rapidez máxima y media de corrientes en el interior del Puerto de Manzanillo, Colima, bajo la acción de los tres principales tsunamis históricos, a partir de tsunamis sintéticos locales. Con una precisión del orden del 20% de error porcentual absoluto medio, como evidencia, se muestran los resultados en la Figura 4.1.



**Figura 4.1 Patrones de rapidez máxima (superior) y media (inferior). En símbolos '+' negros el tsunami local histórico y la línea azul continua es la respuesta de la RNAf. El intervalo de confianza en sombra azul claro**



## Bibliografía

---

- [1] TURBAN, E., ARONSON, J.E., y LIANG, T.P. Decision support systems and intelligent systems. Pearson/Prentice Hall. 2005.
- [2] HAYKIN, Simon S. Neural Networks: A comprehensive foundation. Macmillan College Publishing Company. 1994.
- [3] MANDAL, Subhendu, et. al. Application of Neural Networks in Coastal Engineering – an Overview. International conference of international association for computer methods and advances in geomechanics, (12<sup>a</sup>: Oct. 2008: India).
- [4] DEAN Robert y Dalrymple Robert. Water Wave Mechanics for Engineers and Scientists. Advanced series on ocean engineering. World Scientific. 1991.
- [5] SÁNCHEZ DÉVORA, Antonio, y FARRERAS SANZ, Salvador. Catálogo de Tsunamis (Maremotos) en la Costa Occidental de México. World Data Center A for Solid Earth Geophysics Publication SE-50. 1993.
- [6] VÁSQUEZ LÓPEZ Juan P. Red neuronal feedforward como estimador de patrones de corrientes en el interior del puerto de manzanillo sujeto a la acción de tsunamis. Proyecto de investigación de iniciativa interna IV-17/13. Publicación técnica 406. Instituto Mexicano del Transporte. 2014
- [7] MONTOYA, JOSÉ MIGUEL, FLORES ÁLVAREZ, JUAN ESTÉBAN, CASAS VALENCIA, CINDY, et al. 2013. Estudio en modelo hidráulico para determinar el patrón de velocidades originado por tsunamis locales en el Puerto de Manzanillo, Col. Reporte Técnico VE-12/13. Instituto Mexicano del Transporte.



# Anexo. Rutinas de código

---

## sismosLocales.R

```
# LIBRERIAS Y VALORES INICIALES -----
rm(list= ls())

library(RMySQL) #para conectarse con la BD
library(doMC) #para cómputo paralelo
library(AMORE) #para RNAF
subdirTrabajo<- "~/Dropbox/2014/2014-07-21-VI-27-14/rutinasRv01/"
nCP<- 6 #número de casos de entrenamiento
colObj<- 7 #6 rapMax o 7 para rapMedia
registerDoMC(cores= 4) #4 bueno para Core-i5
system("python 01-preparaBD.py root imt123")

setwd(subdirTrabajo)
source("funcArmaDatos.R")
locales <- armaDatos("datos/locales/")
locales.sinteticos<- armaDatos("datos/sinteticosLocales/")
todo <- rbind(locales, locales.sinteticos)
todo.normalizado<- sapply(todo, function(original) normalizado <- (original -
  mean(original))/sd(original))
locales.normalizado<- head(todo.normalizado, dim(locales)[1])
locales.sinteticos.normalizado<- tail(todo.normalizado,
  dim(locales.sinteticos)[1])
rm(todo, todo.normalizado)

# CONJUNTOS DOMINIO-----
colsP<- rev(list(c(1, 2, 3, 4, 5), c(1, 2), c(1, 2, 5))) #columnas de entrada
colsQ<- rev(list(c(colObj), c(colObj, 1), c(colObj, 1, 2), c(colObj,
  1, 5), c(colObj, 2, 5), c(colObj, 1, 2, 5))) #columnas de salida
LR <- c(0.02, 0.002, 2e-04, 2e-05) #tasas de aprendizaje
MG <- c(0.05, 0.005, 5e-04, 5e-05) #momenta
SS <- c(1000, 9000, 90000) #pasos de entrenamiento
FF <- c(2, 2.25, 2.5, 2.75, 3, 3.25) #factores p/núm. neuronas capa oculta
picked<- 0 #bandera: combinación en proceso
done <- 0 #bandera: combinación ya procesada

# PREPARACIÓN DE BASE DE DATOS -----
base <- "corridas"
usuario <- "userRNAF"
passw<- "imt123"
bd<- dbConnect(MySQL(), user = usuario, password = passw, dbname= base,
  host = "localhost")
```

```
dbSendQuery(bd, "TRUNCATE combinaciones") #borra registros pasados
dbDisconnect(bd)

# CICLOS PARALELIZADOS PARA LLENAR COMBINACIONES EN BD -----
for (cPin colsP) {
  for (cQin colsQ) {
    for (factor in FF) {
      foreach(k = 1:length(LR)) %dopar% {
        for (j in 1:length(MG)) {
          foreach(i = 1:length(SS)) %dopar% {
            bd<- dbConnect(MySQL(), user = usuario,
                          password = passw, dbname= base, host =
                          "localhost")

            lr= LR[k]
            mg = MG[j]
            ss= SS[i]
            orden<- paste("INSERT INTO combinaciones
                          (nCP,colsP,colsQ,hidden,lr,mg,ss,picked,done)
                          VALUES nCP, ',' , paste(as.character(cP),
                          collapse = "-"),
                          ',' , ',' , paste(as.character(cQ), collapse =
                          "-"),
                          ',' , ',' , (floor(length(cP) * factor + 1)),
                          ',' , lr, ',' , mg, ',' , ss, ',' , picked,
                          ',' , done, ')", sep= ",")
            dbSendQuery(bd, orden)
            dbDisconnect(bd) #desconecta la base de datos
          }
        }
      }
    }
  }
}
```

## funcArmaDatos.R

```
armaDatos<- function(ruta, fraccPtosModelo= 1:50) {
  datos <- data.frame()
  x <- list.files(ruta, full.names= TRUE)
  sapply(x, function(y) {
    puntos.numPuntos<- fraccPtosModelo
    y.eta0 <- strsplit(y, "/")[1][2]
    y.eta0 <- as.numeric(substr(y.eta0, start = 6, stop = 9))/100
    puntos.eta0 <- rep(y.eta0, length(fraccPtosModelo))
    puntos.lonLatZ<- read.table("datos/TranspuestasXYZ.dat")
    puntos.lonLatZ<- puntos.lonLatZ[fraccPtosModelo, ]
    d <- read.table(y)
    d <- d[, fraccPtosModelo]
  })
}
```

```

vx<- d[1:59, ]
vy<- d[60:118, ]
rapidez <- sqrt(vx* vx+ vy* vy)
rapidez.Maxima<- sapply(rapidez, max)
rapidez.Media<- sapply(rapidez, mean)
datos.parciales<- cbind(puntos.numPuntos, puntos.eta0,
puntos.lonLatZ, rapidez.Maxima, rapidez.Media)
names(datos.parciales) <- c("numPunto", "eta0", "lon",
"lat", "z", "rapidez.Maxima", "rapidez.Media")
datos <-<- rbind(datos, datos.parciales)
})
datos
}

```

## 01-preparaBD.py

```

importMySQLdbasdb
import sys
con = db.connect('localhost',sys.argv[1], sys.argv[2])
cur = con.cursor()
cur.execute('SHOW DATABASES;')
todas = cur.fetchall()
bandera=0

for base intodas:
    if base[0] == 'corridas':
        bandera=1

ifbandera == 1:
    cur.execute('DROP DATABASE corridas;')
    cur.execute("DROP USER 'userRNAf'@'localhost';")

cur.execute('CREATE DATABASE corridas;')
cur.execute('USE corridas;')
cur.execute('CREATE TABLE combinaciones (nCP INT, colsP CHAR(30), colsQ
CHAR(30), hidden INT, lr DOUBLE, mg DOUBLE, ss DOUBLE, picked INT, done INT,
mape DOUBLE, sigma DOUBLE, lmIs DOUBLE, tf DOUBLE, tff DOUBLE, colObj
INT);')
cur.execute("CREATE USER 'userRNAf'@'localhost' IDENTIFIED BY 'imt123;")
cur.execute("GRANT ALL ON corridas.* TO 'userRNAf'@'localhost;")

if con:
    con.close()

```

## cálculos.R

```
# LIBRERIAS Y VALORES INICIALES-----
library(RMySQL) #para conectarse con la BD
library(doMC) #para cómputo paralelo
library(AMORE) #para RNAf
rm(list= ls())
subdirTrabajo<- "~/Dropbox/2014/2014-07-21-VI-27-14/rutinasRv01/"
colObj<- 7 #6 rapMax o 7 para rapMedia
registerDoMC(cores= 4) #4 bueno para Core-i5
set.seed(12)
setwd(subdirTrabajo)
source("funcArmaDatos.R")
locales <- armaDatos("datos/locales/")
locales.sinteticos<- armaDatos("datos/sinteticosLocales/")
todo <- rbind(locales, locales.sinteticos)
todo.normalizado<- sapply(todo, function(original) normalizado <- (original -
  mean(original))/sd(original))
locales.normalizado<- head(todo.normalizado, dim(locales)[1])
locales.sinteticos.normalizado<- tail(todo.normalizado,
dim(locales.sinteticos)[1])
rm(todo, todo.normalizado)

# PREPARACIÓN DE BASE DE DATOS -----
base <- "corridas"
usuario <- "userRNAf"
passw<- "imt123"

# CICLO PARALELIZADO QUE EJECUTA TODAS LAS COMBINACIONES
# REGISTRAD ----- ARREGLA Y CUENTA CICLOS RESTANTES
t0 <- proc.time() #iniciacronómetro
bd<- dbConnect(MySQL(), user = usuario, password = passw, dbname= base,
  host = "localhost")
Q0 <- "UPDATE combinaciones SET picked=0 WHERE picked=1 AND done=0"
dbGetQuery(bd, Q0) #reparainterrupción
Q1 <- "SELECT * FROM combinaciones WHERE picked=0 AND done=0"
combinacs<- dbGetQuery(bd, Q1) #faltan por procesar
dbDisconnect(bd)

# EJECUTA LOS RESTANTES CICLOS
foreach(k = 1:dim(combinacs)[1], .inorder= FALSE, .combine = rbind) %dopar%
{
  set.seed(12) #homologa semilla aleatoria
  caso <- combinacs[k, ] #cada una de las combinaciones
  bd<- dbConnect(MySQL(), user = usuario, password = passw,
  dbname= base, host = "localhost")
  marcaPicked<- paste("UPDATE combinaciones SET picked = 1 WHERE nCP=",
  caso$nCP, " AND colsP='", caso$colsP, "' AND colsQ='",
```

```

      caso$colsQ, "" AND hidden=", caso$hidden, " AND lr=",
      caso$lr, " AND mg=", caso$mg, " AND ss=", caso$ss,
      sep= "")
dbGetQuery(bd, marcaPicked) #asegura para procesar
t1 <- proc.time() #inicia cronómetro local del caso

# LEE LAS COLUMNAS DE ENTRADA
cP<- eval(parse(text= as.character(paste("c(",
      paste(strsplit(as.character(caso$colsP),
      "-")[[1]], collapse= ","), ")"))))

# LEE LAS COLUMNAS DE SALIDA
cQ<- eval(parse(text= as.character(paste("c(",
      paste(strsplit(as.character(caso$colsQ),
      "-")[[1]], collapse= ","), ")"))))
P <- as.matrix(locales.sinteticos.normalizado[, cP]) #matriz de entrada a
la RNAf
Q <- as.matrix(rbind(as.matrix(locales.normalizado[,
      cQ]), as.matrix(locales.normalizado[, cQ]))) #patrón de
entrenamiento

# ARMADO DE LA RNAf
set.seed(12)
net.start<- newff(n.neurons= c(dim(P)[2], caso$hidden,
      length(cQ)), learning.rate.global= caso$lr, momentum.global=
caso$mg,
      error.criterium= "LMLS", stao= NA, hidden.layer= "tansig",
      output.layer= "purelin", method = "ADAPTgdwm")

# ENTRENAMIENTO
set.seed(12)
result <- train(net.start, P, Q, error.criterium= "LMLS",
      report = TRUE, show.step= caso$ss, n.shows= 3)

# SIMULACIÓN
sim<- sim.MLPnet(result$net, P)
# valores observados (resto de datos)
obs<- Q
# modelo lineal simulados vs observados
fit<- lm(sim[, 1] ~ obs[, 1])
# cálculo del MAPE
mape<- mean(abs((sim[, 1] - obs[, 1])/obs[, 1]))
tf<- round((proc.time() - t1)[[3]], 4) #tiempo de cómputo RNAf
tff<- round((proc.time() - t0)[[3]], 4) #tiempo total del programa

# SALIDA DE RESULTADO A BD
marcaDone= paste("UPDATE combinaciones SET done = 1, mape=",
      mape, ", sigma=", round(summary(fit)$sigma, 4), ", lmls=",
      round(result$merror[3], 4), ", tf=", tf, ", tff=",
      tff, ", colobj=", colobj, " WHERE nCP=", caso$nCP,
      " AND colsP='", caso$colsP, "' AND colsQ='", caso$colsQ,

```

```
    " AND hidden=", caso$hidden, " AND lr=", caso$lr,
    " AND mg=", caso$mg, " AND ss=", caso$ss, " AND picked=",
    1, sep= "")
    dbGetQuery(bd, marcaDone) #marca la combinación de 'procesada'

# DESCONECTA DE LA BD
dbDisconnect(bd)
}
```

## resultados.R

```
rm(list= ls())
library(RMySQL) #para conectarse con la BD
library(AMORE) #para RNAf
subdirTrabajo<- "~/Dropbox/2014/2014-07-21-VI-28-14/rutinasRv01/"
colObj<- 7 #6 rapMax o 7 para rapMedia
nombreFig1 <- "RNAf-rapMedia.dot"
nombreFig2 <- "1mRapMedia.png"
nombreFig3 <- "RapMedia.png"
yLabelFig3 <- "rapidez media normalizada (u.a.)"
source("funcArmaDatos.R")
locales <- armaDatos("datos/sinteticosBrecha/")
locales.sinteticos<- armaDatos("datos/sinteticosLocales/")
todo <- rbind(locales, locales.sinteticos)
todo.normalizado<- sapply(todo, function(original) normalizado <- (original -
mean(original))/sd(original))
locales.normalizado<- head(todo.normalizado, dim(locales)[1])
locales.sinteticos.normalizado<- tail(todo.normalizado,
dim(locales.sinteticos)[1])
rm(todo, todo.normalizado)

# PREPARACIÓN DE BASE DE DATOS -----
base <- "corridas"
usuario <- "userRNAf"
passw<- "imt123"

# RECUPERA LA RNAf OPTIMA -----
bd<- dbConnect(MySQL(), user = usuario, password = passw, dbname= base,host=
"localhost")
q0 <- "SELECT * FROM combinaciones WHERE mape IN (SELECT MIN(mape) FROM
combinaciones)"
pmin<- dbGetQuery(bd, q0) #recupera parámetros de la RNAf optima
dbDisconnect(bd)
cP<- eval(parse(text = as.character(paste("c(",
paste(strsplit(as.character(pmin$colsP),
"-")[1], collapse = ","), ")"))))
cQ<- eval(parse(text = as.character(paste("c(",
paste(strsplit(as.character(pmin$colsQ),
"-")[1], collapse= ","), ")"))))
P <- as.matrix(locales.sinteticos.normalizado[, cP]) #matriz de entrada a la
RNAf
Q <- as.matrix(rbind(as.matrix(locales.normalizado[, cQ]),
as.matrix(locales.normalizado[,
cQ]))) #patrón de entrenamiento
```

```

# armado de la RNAf
net.start<- newff(n.neurons= c(dim(P)[2], pmin$hidden, length(cQ)),
learning.rate.global= pmin$lr, momentum.global= pmin$mg,
error.criterion= "LMLS", Stao= NA, hidden.layer= "tansig",
output.layer= "purelin", method = "ADAPTgdwm")

# entrenamiento
set.seed(12)
result <- train(net.start, P, Q, error.criterion= "LMLS", report = TRUE,
show.step= pmin$ss, n.shows= 3)

# simulación
set.seed(12)
sim<- sim.MLPnet(result$net, P)
obs<- Q
fit<- lm(sim[, 1] ~ obs[, 1])# modelo lineal simulados vs observados
mape<- mean(abs((sim[, 1] - obs[, 1])/obs[, 1])) #MAPE
rmse<- round(sqrt(mean(resid(fit)^2)), 4) #RMSE

# GRAFICAS DE LA RNAFOPTIMA Y CALIDAD DE PREDICC. -----
graphviz.MLPnet(net = net.start, filename = nombreFig1)

png(filename = nombreFig2, width = 8, height = 4.944, units = "in",res= 600)
layout(matrix(c(1:3), 1, 3, byrow= TRUE))
plot(obs[, 1], sim[, 1], main = paste("MAPE=", round(mape, 4),
sep= ""), ylab= "RNAf", xlab= "observados")
deltaOrdenada<- summary(fit)$coefficients[1] +
summary(fit)$coefficients[3]
abline(deltaOrdenada, 1)
abline(-deltaOrdenada, 1)
abline(0, 1)
hist(fit$residuals, probability = T, main = "", xlab= "residuales")
lines(density(fit$residuals), lwd= 2)
plot(fit, which = 2)
dev.off()

png(filename = nombreFig3, width = 8, height = 4.944, units = "in",res= 600)
layout(matrix(c(1:3), 1, 3, byrow= TRUE))
graficaParcial<- function(tramo, leyendaY, leyendaX, etaVal,
leyendaMain) {
mape<- mean(abs((sim[tramo, 1] - obs[tramo, 1])/obs[tramo,1])) #MAPE
low <- sim[tramo, 1] * (1 - mape)
up <- sim[tramo, 1] * (1 + mape)
plot(obs[tramo, 1], type= "n", ylab= leyendaY, xlab= leyendaX,
ylim= c(min(min(obs[, 1], min(sim[, 1]))), max(max(obs[,1],
max(sim[, 1])))), main = bquote(eta[0] == .(paste(etaval,
"m", "\t", leyendaMain))))
polygon(c((1:50), rev(1:50)), c(low, rev(up)), col = "lightblue",
border = FALSE)
points(obs[tramo, 1], lwd= 1, col = "black", type = "b",pch= "+")
lines(sim[tramo, 1], lwd= 2, col = "blue")
grid()

```

```
}  
graficaParcial(1:50, yLabelFig3, "", unique(locales$eta0)[1], "Sintético  
1")  
graficaParcial(51:100, "", "Puntos de modelación",  
  unique(locales$eta0)[2], "Sintético 2")  
graficaParcial(101:150, "", "", unique(locales$eta0)[3], "Sintético 3")  
dev.off()  
rm(list = ls())
```



Carretera Querétaro-Galindo km 12+000  
CP 76700, Sanfandila  
Pedro Escobedo, Querétaro, México  
Tel +52 (442) 216 9777 ext. 2610  
Fax +52 (442) 216 9671

[publicaciones@imt.mx](mailto:publicaciones@imt.mx)

<http://www.imt.mx/>