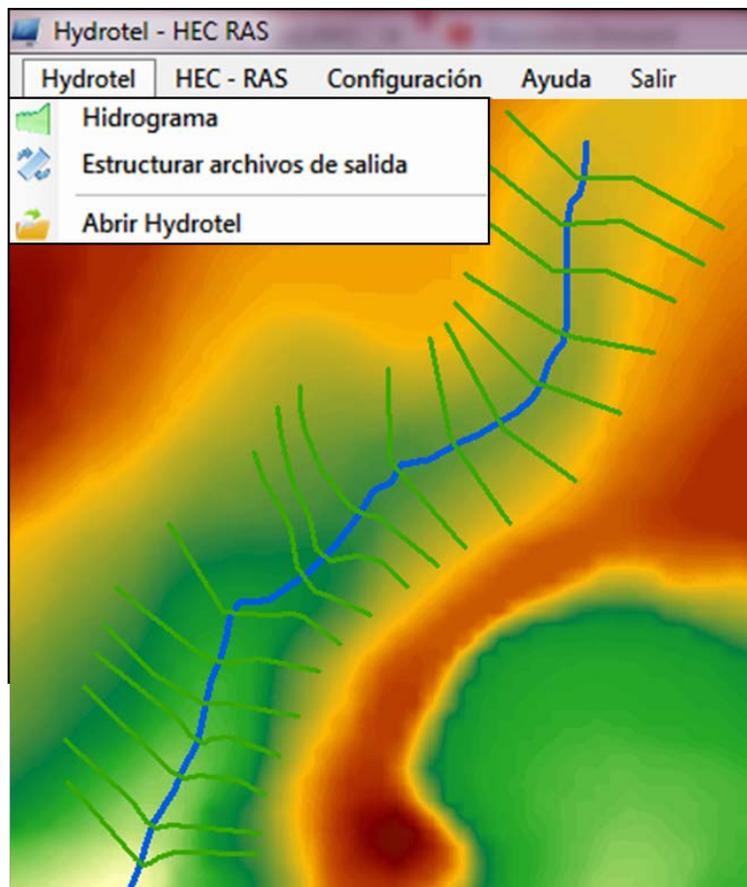


Coordinación de Hidrología
Subcoordinación de Hidrología y Mecánica de Ríos

Proyecto TH1213.2



**Nuevas tecnologías para prevención y análisis de
inundaciones**

Informe Final

Diciembre de 2012

Índice

Introducción	7
1. Modelación hidrológica de la cuenca del río Conchos	11
1.1 Recopilación y análisis de información	11
1.1.1 Información geográfica.....	11
1.1.2 Modelo digital de elevaciones.....	11
1.1.3 Parteaguas y subcuencas.....	12
1.1.4 Red de drenaje	12
1.1.5 Uso de suelo.....	13
1.1.6 Edafología	14
1.2 Series de tiempo	14
1.2.1 Estaciones hidrométricas	15
1.2.2 Estaciones meteorológicas.....	15
1.3 Modelación hidrológica	18
1.4 Proyecto Physitel	19
1.4.1 Importación del modelo digital de elevación	20
1.4.2 Importación y edición de la red vectorial de corrientes impuesta	20
1.4.3 Determinación y edición de la red matricial de corrientes impuesta	21
1.4.4 Determinación y edición del modelo digital de elevación modificado	22
1.4.5 Determinación y edición de pendientes.....	23
1.4.6 Determinación y edición de las direcciones de flujo	24
1.4.7 Determinación del parteaguas de la cuenca	25
1.4.8 Determinación de la red de corrientes modelada.....	26
1.4.9 Determinación y edición de las unidades hidrológicas	27
1.4.10 Importación y edición de los usos de suelo	28
1.4.11 Importación y edición de los tipos de suelo.....	29
1.4.12 Exportación del proyecto <i>Physitel</i> al formato <i>Hydrotel</i>	30
1.5 Proyecto Hydrotel	31
1.5.1 Importación de archivos Physitel.....	31

1.5.2	Definición de archivos fisiográficos	33
1.5.3	Importación de archivos hidrométricos y meteorológicos	33
1.5.4	Creación de grupos de UHRH	36
1.5.5	Definición de gradientes temperatura	37
1.5.6	Generación de hidrogramas geomorfológicos	39
1.5.7	Calibración del modelo hidrológico	39
1.5.8	Simulación histórica del modelo hidrológico	46
1.5.9	Pronóstico de escurrimientos	50
2.	Modelación de la cuenca del río Yautepec	51
2.1	Conceptualización	51
2.1.1	Modelo hidrológico distribuido	51
2.1.2	Modelo hidráulico	51
2.1.3	Discretización temporal	52
2.1.4	Discretización espacial	52
2.1.5	Desarrollo de la interfaz de Usuario	52
2.2	Recopilación y análisis de información	53
2.2.1	Información geográfica	53
2.2.2	Modelo digital de elevaciones	53
2.2.3	Parteaguas y subcuencas	54
2.2.4	Red de drenaje	55
2.2.5	Uso de suelo	57
2.2.6	Edafología	58
2.3	Series de tiempo	59
2.3.1	Estaciones hidrométricas	59
2.3.2	Estaciones meteorológicas	60
2.4	Proyecto Physitel	62
2.4.1	Importación del modelo digital de elevación	62
2.4.2	Determinación y edición de pendientes	63
2.4.3	Importación y edición de la red vectorial de corrientes impuesta	64
2.4.4	Determinación y edición de la red matricial de corrientes impuesta	64
2.4.5	Determinación y edición del modelo digital de elevación modificado	65
2.4.6	Determinación y edición de las direcciones de flujo	66
2.4.7	Determinación del parteaguas de la cuenca	67
2.4.8	Determinación de la red de corrientes modelada	68

2.4.9	Determinación y edición de las unidades hidrológicas	70
2.4.10	Importación y edición de los usos de suelo	71
2.4.11	Importación y edición de los tipos de suelo	72
2.4.12	Exportación del proyecto <i>Physitel</i> al formato <i>Hydrotel</i> ...	73
2.5	Proyecto Hydrotel	73
2.5.1	Importación de archivos Physitel	73
2.5.2	Definición de archivos fisiográficos	74
2.5.3	Importación de archivos hidrométricos y meteorológicos	75
2.5.4	Creación de grupos de UHRH	78
2.5.5	Definición de gradientes temperatura	78
2.5.6	Generación de hidrogramas geomorfológicos	79
2.5.7	Calibración del modelo hidrológico	80
2.5.8	Simulación de tramos en confluencia del río Yautepec	82
2.6	Proyecto HEC-RAS	83
2.6.1	Recopilación y análisis de información	83
2.6.2	Preparación de datos	85
2.6.3	Modelo HEC-RAS	85
2.6.4	Mapeo de la planicie de inundación	86
3.	Interfaz de usuario.	87
3.1	Descripción de ventanas del mapa conceptual	88
3.2	Modelo hidrológico Hydrotel	89
3.3	Modelo hidráulico HEC-RAS	95
3.4	Configuración de resultados	97
4.	Resultados	¡Error! Marcador no definido.
4.1	Modelo hidrológico	¡Error! Marcador no definido.
4.2	Modelo hidráulico	¡Error! Marcador no definido.
4.3	Modelos coordinados	¡Error! Marcador no definido.
	Conclusiones y recomendaciones	99
	Referencias	101
	Anexos	102
	Introducción	108
1.	Mapa conceptual de la interfaz de usuario	109
2.	Descripción de ventanas del mapa conceptual	110
2.1	Modelo hidrológico <i>Hydrotel</i>	110
2.1.1	Hidrograma	111
2.1.2	Estructurar archivos de salida	113

2.1.3 Abrir <i>Hydrotel</i>	115
2.2 Modelo hidráulico <i>HEC-RAS</i>	116
2.2.1 Gasto máximo	117
2.2.2 Abrir <i>HEC-RAS</i>	119
3. Configuración del sistema	121
Principal.cs	124
Configuracion.cs	127
Cálculo de gasto máximo qMax.cs	130
DatosDiarios.cs	138
Hidrograma.cs	147



Participantes

Jorge H. Salgado Rabadán Jefe de proyecto

Alberto Güitrón de los Reyes

Iván Rivas Acosta

Gabriela Colorado Ruiz

Margarita Preciado Jiménez

José A. Bravo Jácome

Alexis Cervantes Carretero

Introducción

Las inundaciones son fenómenos naturales que son considerados fenómenos de mayor impacto en el ámbito mundial se convierten en peligros cuando los espacios ocupados por las poblaciones abarcan las llanuras de inundación naturales de un río y, por consecuencia, son afectadas por la acumulación de agua en extensiones territoriales densamente pobladas.

De acuerdo con la base internacional de datos de desastres elaborada por el CRED (OFDA-CRED Database), las avenidas han recibido mayor atención en el mundo debido a su repercusión sobre la población, debido a la ocurrencia de diversos desastres asociados a este fenómeno natural.

Nuestro país frecuentemente es afectado por precipitaciones producidas por actividad ciclónica y tormentas tropicales debido a las condiciones geográficas y fisiográficas, la ocurrencia de este fenómeno es durante el verano y otoño, y por los frentes fríos en invierno, que ocasionan lluvias abundantes en diferentes estados, tales como Chiapas, Puebla, Tabasco, Veracruz, Yucatán, entre otros.

Al respecto, se han realizado diversas investigaciones que han abordado la problemática de las inundaciones en distintas regiones de nuestro país considerando enfoques diversos; entre ellos, cabe destacar el trabajo de Delgado e *Isuhuaylas (2001)*, quienes emplearon modelos hidráulicos para entender la dinámica de las inundaciones en la subcuenca del río Tejalpa, en el Estado de México, así como *Hudson y Colditz (2003)* combinaron criterios geomorfológicos, de laboratorio y percepción remota para delimitar un valle aluvial en la cuenca del río Pánuco.

Existen diversos métodos para el pronóstico de avenidas que se han aplicado a la prevención de inundaciones en el país, pero la falta de datos suficientes y apropiados frecuentemente conduce a errores significativos. Sin embargo, la base para aplicar las medidas de apoyo que minimizan los riesgos es conocer las zonas afectadas y la magnitud del evento hidrometeorológico durante una inundación con el mayor detalle posible con los modelos de simulación y sistemas de información geográfica, que son las herramientas más importantes para lograr un manejo del riesgo y los daños por eventos extremos en zonas de interés. Por lo tanto, para la determinación de la zona afectada, actualmente la herramienta principal son los mapas de riesgo.

Por otra parte, se estima que el cambio climático global modificará el régimen hidrológico significativamente. Aunque en México los cambios en el régimen anual de precipitación apuntarán a la baja de acuerdo con las estimaciones publicadas hasta el momento, los fenómenos extremos podrán no seguir la misma tendencia. Por tanto, es necesario revisar los

parámetros de diseño de las obras hidráulicas existentes y estimar cómo se modificarán por los efectos del cambio climático.

Para abordar esta problemática, se desarrolló un algoritmo que permite la comunicación de dos modelos, uno que transforma la lluvia en escurrimiento y otro que transita el escurrimiento en una corriente natural. Con los resultados del caudal transitado y asociado a un modelo digital de elevación, se tienen elementos para definir el área de inundación con la precisión requerida.

Estas actividades se desarrollaron en el Instituto Mexicano de Tecnología del Agua, en el año 2012 con recursos propios y disponibilidad inicial del año anterior, bajo el nombre de "*Elaboración de un Atlas nacional de riesgo de inundaciones*", con clave de proyecto TH1213.2.

Este Informe Final está organizado en cuatro capítulos, en el capítulo uno se presenta la modelación hidrológica del río Conchos, en el capítulo dos se describe la modelación del río Yautepec que consiste en un modelo hidrológico y uno hidráulico, en el capítulo tres se describe el desarrollo de la interfaz de usuario y en el capítulo cuatro se presentan los resultados obtenidos tanto de las modelaciones hidrológicas como de la modelación hidráulica. A continuación se enuncian los objetivos generales y específicos del proyecto.

Objetivos generales

Construir un modelo hidrológico distribuido integrado a un modelo de simulación de tránsito de avenidas en cauces que permita definir áreas de inundación y aplicarlo a la cuenca del río Conchos, en la parte de aguas abajo de la presa Luis L. León.

Objetivos específicos

Construir un modelo hidrológico distribuido (modelo *Physitel/Hydrotel*) que calcule los escurrimientos en la cuenca alta del río Conchos, acoplado con un modelo de tránsito de avenidas (modelo HEC-RAS), y con base en estos modelos construir una interfaz de usuario que permita operar de los dos modelos de simulación.

Metodología

Se realizó la modelación hidrológica distribuida del río Conchos y del río Yautepec. Con la primera modelación pretende medir el impacto de la reducción en la precipitación en los escurrimientos en la cuenca, utilizando pronósticos de lluvia de seis meses. Con esta modelación de escurrimientos, se aportan elementos para la toma de decisiones, en el ámbito de la gestión de los recursos hidráulicos. Con la modelación hidrológica del río Yautepec se generaron hidrogramas en un punto crítico de la cuenca debido a las inundaciones que se han suscitado en los últimos años.

Partiendo del punto crítico identificado en la cuenca del río Yautepec, se desarrolló un modelo hidrológico que permite transitar el hidrograma resultante del modelo hidrológico.

Ambos modelos los hidrológicos y el hidráulico fueron desarrollados en plataformas diferentes, y para subsanar esta limitante se desarrolló una interfaz de usuario para intercambiar información entre ambas plataformas, facilitando la operatividad de dichos modelos.

Los resultados de la modelación hidráulica son cruzados con un modelo digital de elevación lo cual permite definir el contorno de la superficie inundable.

Contexto de la cuenca del río Conchos

La cuenca del río Conchos es uno de los sistemas ribereños más importantes de todo el norte de México, forma el área de aportación más grande de las corrientes mexicanas que descargan al río Bravo. El río Conchos pertenece a la vertiente del golfo de México, tiene una superficie de escurrimiento de 65,770 kilómetros cuadrados, con una Temperatura media anual 8-18°C y precipitación total anual que oscila entre los 300-1 000 mm en la parte alta de la cuenca.

La cuenca de estudio comprende parte del estado de Chihuahua y una pequeña parte del estado de Durango, ya que el río Conchos nace al suroeste del estado de Chihuahua con dos afluentes principales en este estado, el río San Pedro y el río Chuviscar, el otro afluente principal es el río Florido, que nace en el estado de Durango.

En la zona de estudio existen varias presas, sin embargo, son siete los principales almacenamientos destinados principalmente al riego agrícola, demanda urbana y control de avenidas.

Nombre	Capacidad total referida al NAMO (hm³)
La Boquilla	2,903.36
Francisco I. Madero	348.00
Chihuahua	25.84
El Rejón	6.60
Luis L. León	356.00
Federalismo Mexicano	255.43
Pico del Águila	51.11

Contexto de la cuenca del río Yautepec

El río Yautepec recorre la zona central del Estado de Morelos y es de las principales corrientes de agua que existen en la zona. Su cauce inicia, dentro del Estado, en la zona norte, bajo el nombre de Jerusalén y continúa hacia el sur hasta encontrarse con el Río Amacuzac. En su trayectoria, el río Yautepec alcanza, bajo este nombre, los municipios de Yautepec, Tlaltizapán, Talquitenango y Jojutla.

La cuenca hidrológica del río Yautepec cubre 1,094 km² hasta la confluencia con el río Amacuzac. En esta cuenca se localiza la localidad de Yautepec Morelos, que tiene una población de 84,405 habitantes. La precipitación promedio anual es de 946 mm de acuerdo con el Servicio Meteorológico Nacional en el periodo 1941-2001, El uso consuntivo principal en la región es agrícola, múltiples unidades de riego.

Las afectaciones que ha sufrido la localidad de Yautepec debido a las inundaciones en ocurridas en los últimos años, es un factor fundamental para el análisis hidrológico e hidráulico de la cuenca. En este sentido, dentro de la actividad de identificar zonas de riesgo de inundaciones el IMTA propuso abordar la problemática mediante la realización de dos modelos coordinados de simulación de cuenca de tipo hidrológico distribuido y de tránsito de avenidas.

Modelos de simulación coordinados

El modelo hidrológico distribuido de la cuenca del río Yautepec (1,094 km²) utiliza una base de datos desarrollada en *Physitel* en donde se integra información referente a las características fisiográficas de la cuenca, la cual es exportada a *hydrotel* en donde se calcula el escurrimiento con base en unidades hidrológicas previamente determinadas. Aprovecha al máximo la información fisiográfica y meteorológica disponible. Fue calibrado para varias subcuencas con series hidrométricas.

El modelo hidrológico aporta insumos al modelo de tránsito de avenidas. Permite generar series de variables con intervalo diario y horizonte multianual en cualquier punto no aforado de la cuenca. El modelo de tránsito de avenidas es una herramienta que permite determinar el flujo del agua en el tramo de interés de río para identificar la zona de inundación.

El modelo hidrológico y el modelo de tránsito de avenidas con comunicados a través de una interfaz construida ex profeso para facilitar la operación de los modelos. La interfaz tiene la funcionalidad de activar los modelos de simulación y administrar la información de entrada y salida de ambos modelos, los resultados del modelo hidrológico son el insumo del modelo de tránsito de avenidas.

Los resultados de la simulación del modelo de tránsito de avenidas son agregados espacialmente, para definir el contorno del área afectada por la inundación.

1. Modelación hidrológica de la cuenca del río Conchos

1.1 Recopilación y análisis de información

La recopilación de información consistió en formar una base de datos sobre rasgos geográficos y series de tiempo de la cuenca del río Conchos. La información recopilada proviene de diferentes fuentes, como se indica en cada apartado de este capítulo, en los que se muestra en síntesis la información recopilada para el proyecto. Los anexos 1 y 2 de este informe contienen la información detallada.

1.1.1 Información geográfica

La base de datos geográfica contiene rasgos vectoriales y matriciales en la proyección Cónica Conforme de Lambert, *Datum* WGS 84.

1.1.2 Modelo digital de elevaciones

El Modelo Digital de Elevaciones (MDE) para la cuenca proviene del *Shuttle Radar Topography Mission* (SRTM) versión *finished* producido por el *United States Geological Survey* (USGS). El MDE tiene una resolución de celda de tres segundos lo que equivale a 90 metros (Fig. 1.1). En las zonas donde se muestran huecos de información se utilizó el Modelo Continuo de Elevaciones (MCE) de INEGI.

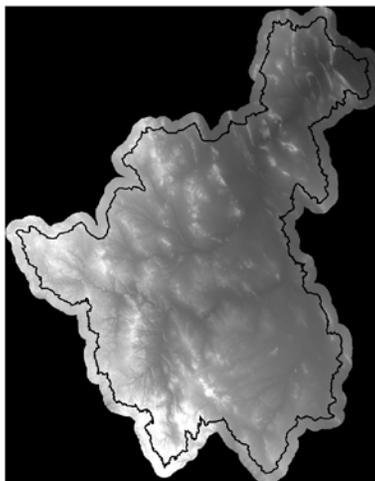


Figura 1.1 Modelo Digital de Elevaciones del SRTM de la cuenca del río Conchos

El MDE se utilizó en la generación de la fisiografía de la cuenca en *Physitel*. Comprende en su totalidad el área de la cuenca junto con un margen de 10 km a partir del parteaguas generado por el *Physitel* para la cuenca.

1.1.3 Parteaguas y subcuencas

De INEGI se recopilaron el mapa en formato shp de las subcuencas 1:250,000 (Fig. 1.2).

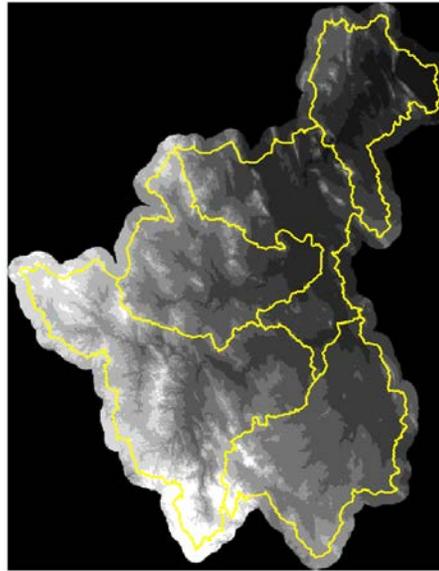


Figura 1.2 Subcuencas INEGI 1:250,000

1.1.4 Red de drenaje

La red de drenaje para la cuenca del río Conchos (Fig. 1.3) es producto de cartas vectoriales escala 1:50,000 del *INEGI*, en las zonas sin información se complementó con la red de corrientes 1:250,000.

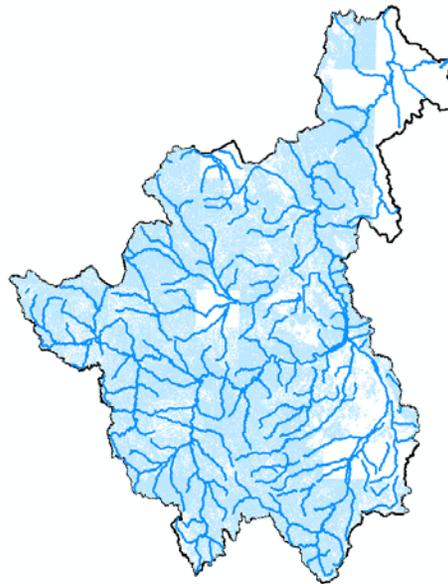


Figura 1.3 Red hidrográfica vectorial escala 1:50,000 de la cuenca del río Conchos

1.1.5 Uso de suelo

Se obtuvo el mapa en formato shp de la cobertura de uso de suelo de la serie IV de INEGI (Fig. 1.4).

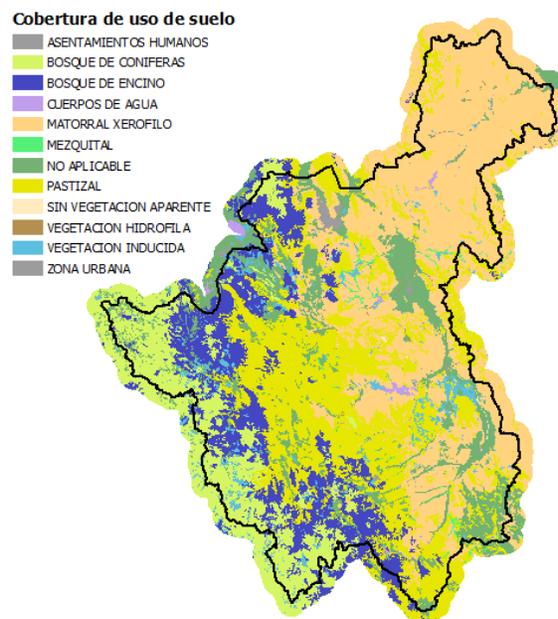


Figura 1.4 Uso de suelo de la cuenca de río Conchos según el sistema de clasificación de cobertura vegetal ajustados por el INE

1.1.6 Edafología

La edafología de la cuenca del río Conchos se obtuvo a partir del mosaico de las cartas edafológicas producido por el INEGI escala 1:250,000. Los tipos de suelo se clasificaron siguiendo los criterios establecidos por la *FAO/UNESCO* en 1968 en el *World Reference Base (WRB)* y modificado por DETENAL en 1970. En México, el sistema de clasificación consta de 25 grandes grupos de clases deducidas a partir de la génesis de los suelos. Los grupos se expanden a un segundo nivel en el que al nombre se les añade un adjetivo para hacer más precisa la clasificación. Las cartas también constan de información sobre la fase física, química y textural ésta última se reduce a establecerla como gruesa media y fina. En la figura 1.5 se muestra la textura del suelo de la cuenca.

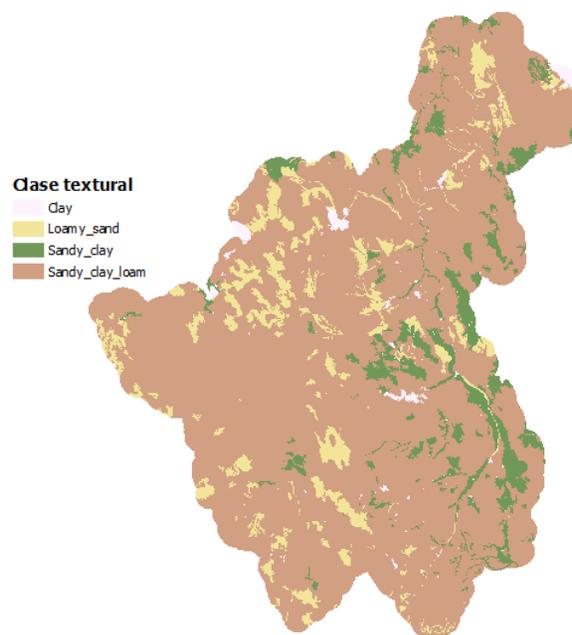


Figura 1.5 Capa de textura de suelo para la cuenca del río Conchos. INEGI seri II 2002-2007

1.2 Series de tiempo

Se prepararon las series históricas completas de temperatura mínima, temperatura máxima y precipitación diarias en formato legible por *Hydrotel*. Asimismo, se prepararon las series históricas completas de escurrimiento diario en tres estaciones de control ubicadas en o cerca de la salida de las subcuencas.

1.2.1 Estaciones hidrométricas

Se identificaron 27 estaciones hidrométricas en la cuenca del Río Conchos, todas se ubicaron geográficamente (Fig. 1.6). Se seleccionaron 7 estaciones de calibración para el modelo hidrológico.

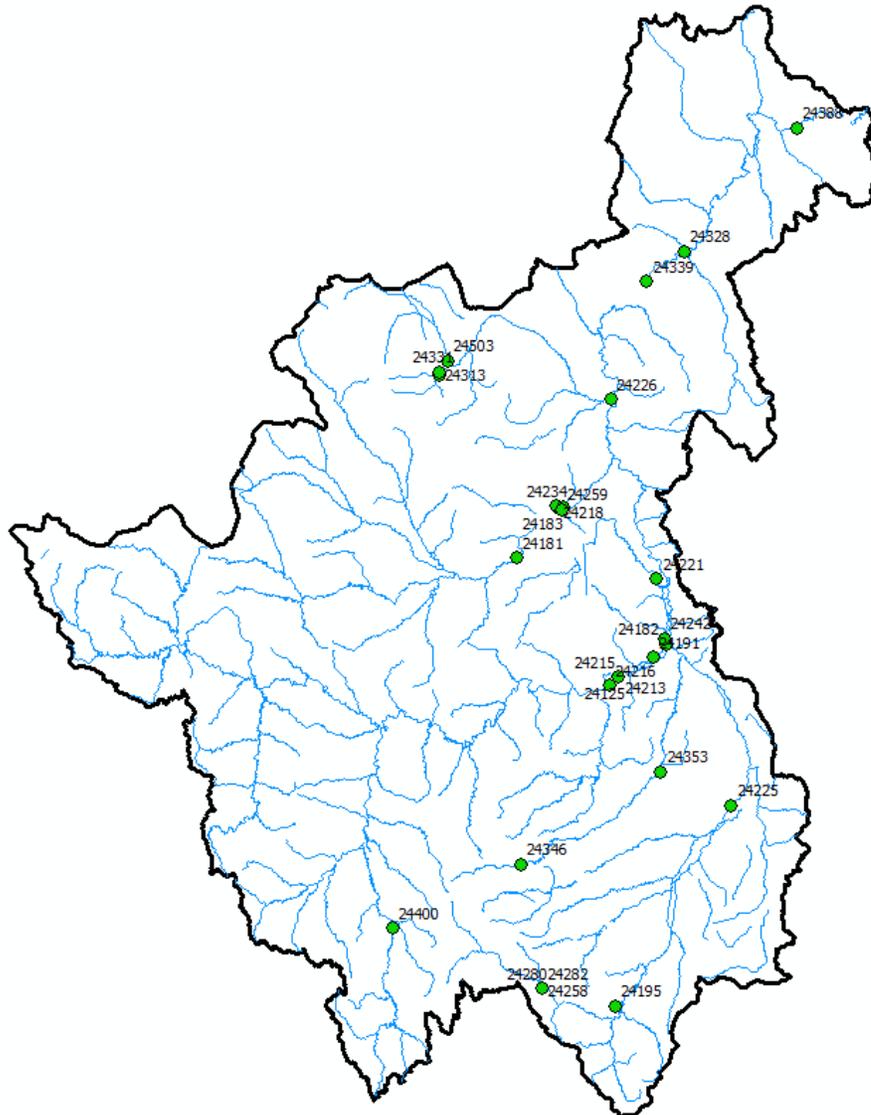


Figura 1.6 Ubicación de las estaciones hidrométricas en la cuenca del río Conchos

1.2.2 Estaciones meteorológicas

De un total de 125 estaciones meteorológicas (ERIC III) que comprenden el área de estudio, con margen de 10 km a partir del parteaguas (Fig. 1.7).

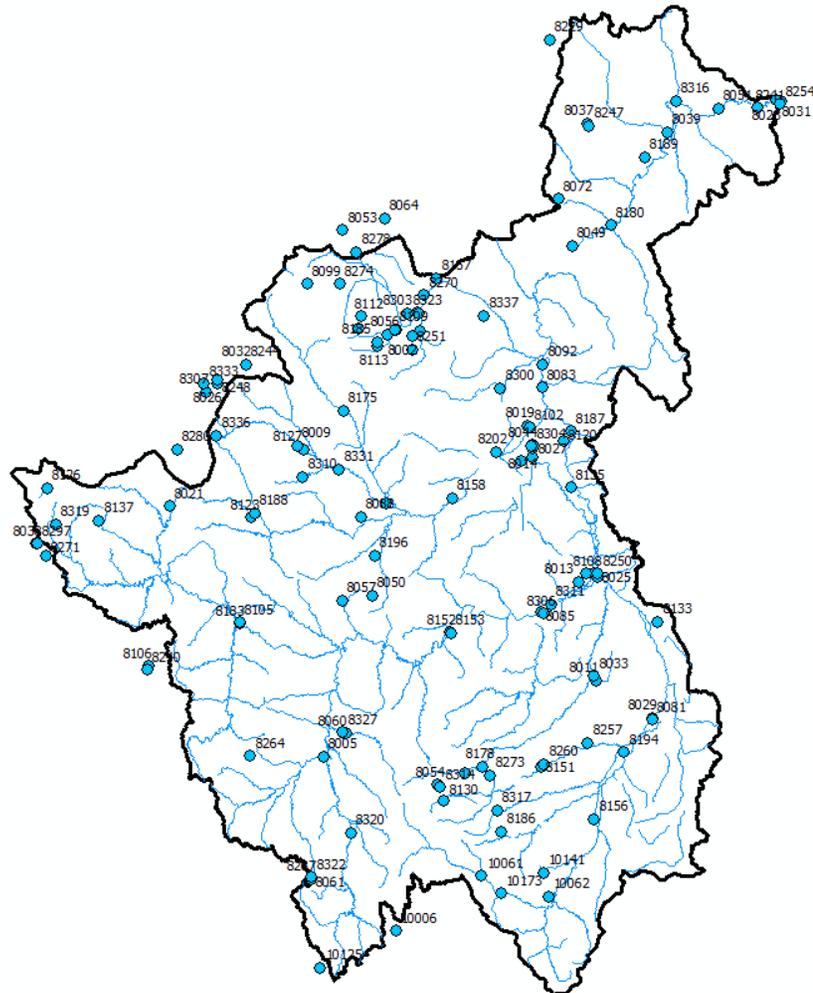


Figura 1.7 Cobertura de estaciones meteorológicas

Se hizo un análisis de las estaciones climatológicas, y se identificó que la mayor densidad de registros se encontró en el periodo de 1980-1984 y la tendencia en cuanto a número de estaciones va sensiblemente a la baja y la cantidad de registros en cada estación su variación se mantiene constante (Fig. 1.8 y 1.9).

Meteorología

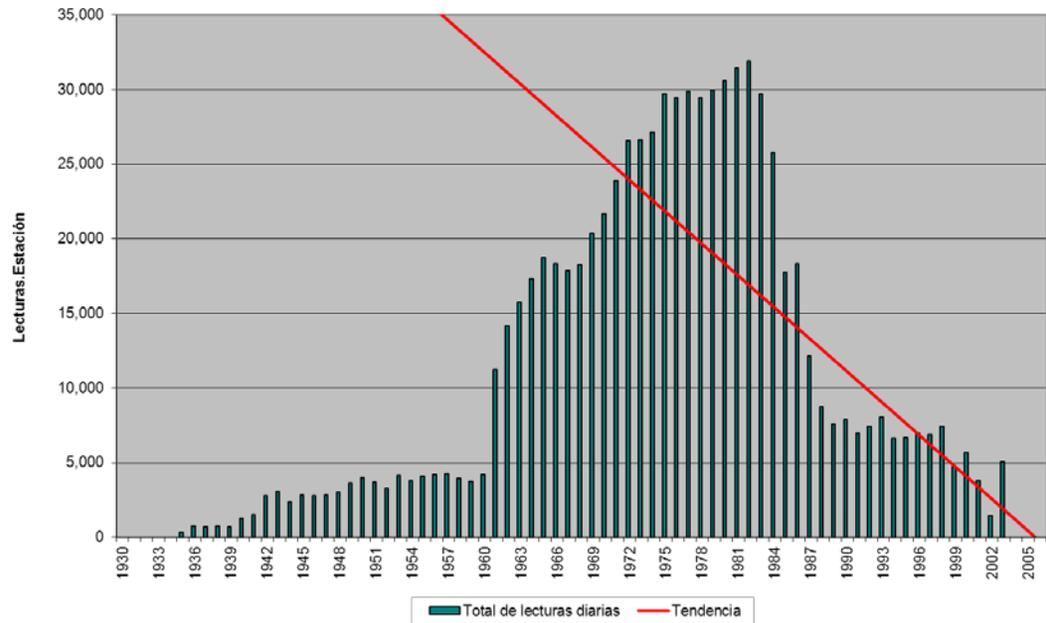


Figura 1.8 Análisis de cantidad de estaciones meteorológicas

Meteorología

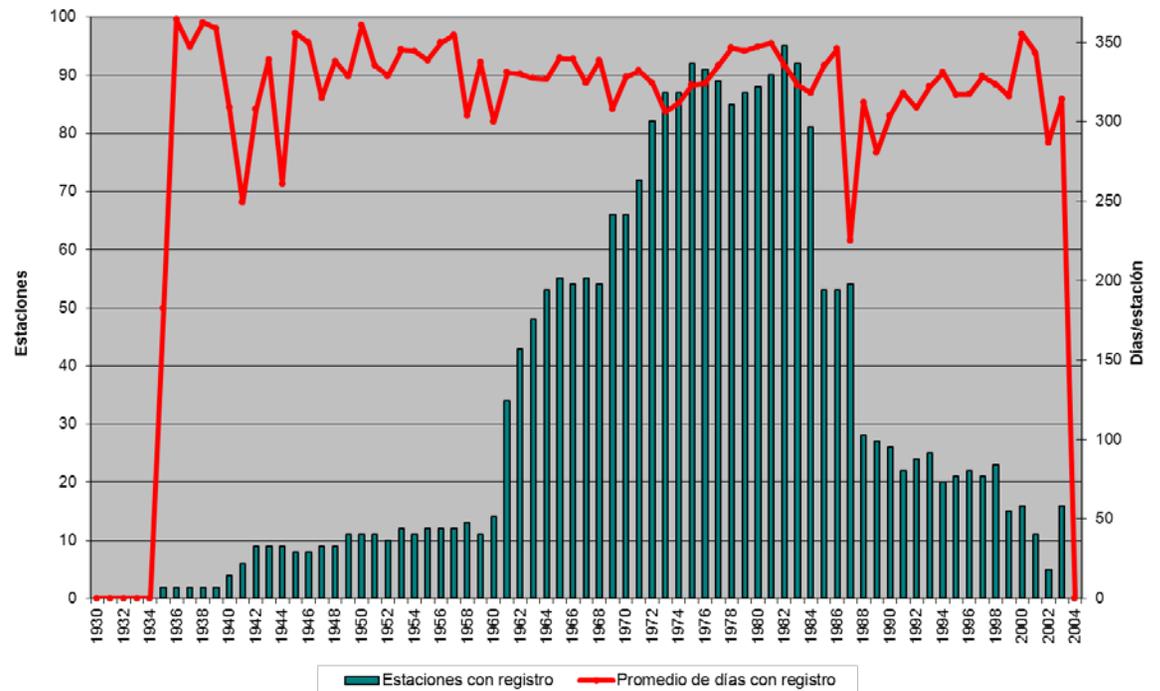


Figura 1.9 Análisis de registros en estaciones meteorológicas

1.3 Modelación hidrológica

El modelo hidrológico distribuido fue desarrollado en la plataforma *Hydrotel*. Simula a intervalo diario el escurrimiento directo, el escurrimiento retardado y el escurrimiento base en cientos de UHRH. La fisiografía de la cuenca se procesa previamente mediante la aplicación *Physitel*.

El modelo hidrológico considera para cada UHRH, mediante seis procesos de simulación, los datos fisiográficos, datos meteorológicos diarios y parámetros de procesos y de calibración. Cuatro de los procesos simulan el flujo vertical del agua en el suelo y los otros dos el flujo horizontal en superficie y cauces (Fig. 1.10).

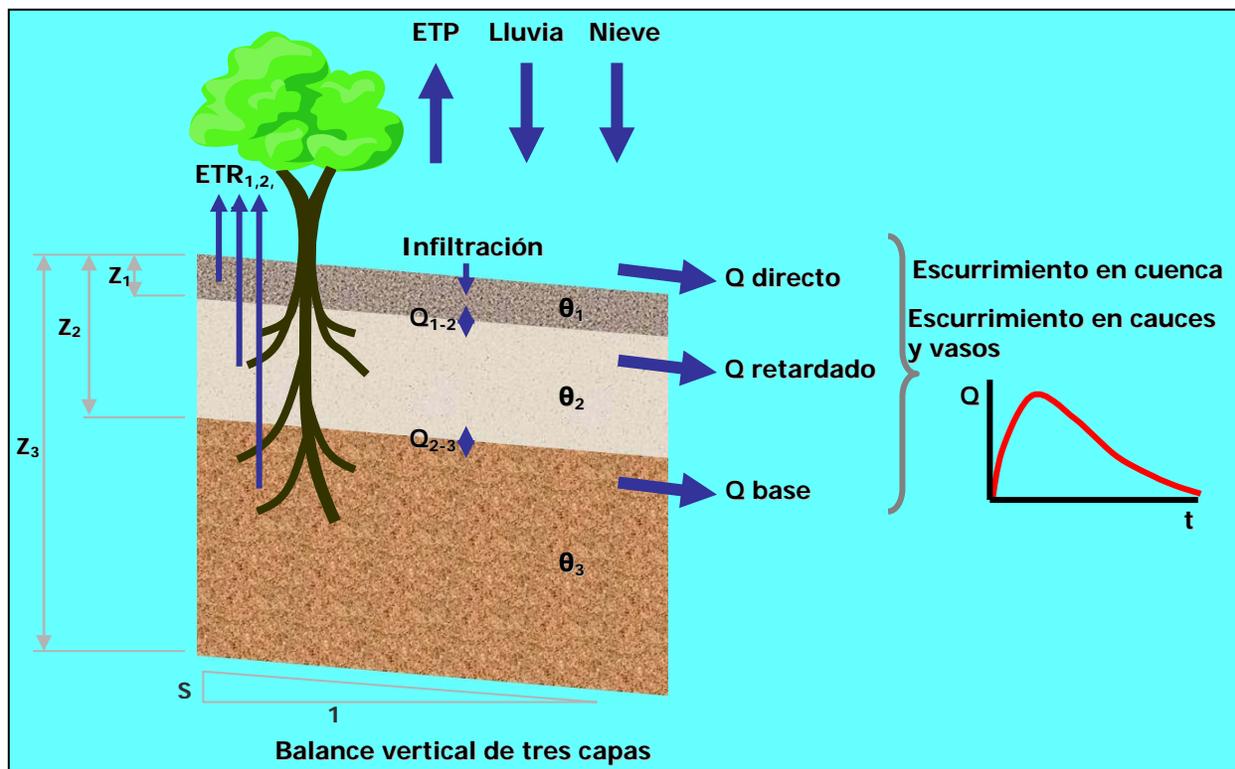


Figura 1.10 Procesos de simulación Hydrotel

Para cada proceso existen algoritmos alternativos que se seleccionan en función de los datos disponibles. En la tabla 1.1 se muestran en *negritas* los algoritmos aplicados.

Procesos	Algoritmos alternativos	Parámetros de calibración
Interpolación de precipitación y de temperaturas máxima y mínima del día	<ul style="list-style-type: none"> Ponderación de las tres estaciones más cercanas Polígonos de <i>Thiessen</i> 	<ul style="list-style-type: none"> Umbral de temperatura del aire para separación de precipitación líquida y sólida

Procesos	Algoritmos alternativos	Parámetros de calibración
Acumulación y fusión de nieve	<ul style="list-style-type: none"> • Enfoque mixto grados-día y balance de energía 	<ul style="list-style-type: none"> • Factores de fusión para tres clases de uso de suelo • Umbrales de temperatura del aire para tres clases de uso de suelo • Tasa de fusión en interfaz nieve- suelo • Coeficiente de compactación
Evapotranspiración potencial	<ul style="list-style-type: none"> • <i>HydroQuebec</i> • <i>Thornthwaite</i> • <i>Penman Monteith</i> • <i>Priestley Taylor</i> • <i>Linacre</i> 	<ul style="list-style-type: none"> • Factor multiplicativo de evapotranspiración (ETP)
Balance vertical de agua	<ul style="list-style-type: none"> • Balance vertical de tres capas • CEQUEAU 	<ul style="list-style-type: none"> • Profundidad de límites inferiores de tres capas de suelo • Coeficiente de recesión
Flujo en la superficie de UHRH	<ul style="list-style-type: none"> • Ecuación de onda cinemática 	<ul style="list-style-type: none"> • Coeficiente de rugosidad de <i>Manning</i> para tres clases de uso de suelo y agua
Tránsito en los cauces y vasos	<ul style="list-style-type: none"> • Ecuación de onda cinemática • Ecuación de onda difusiva 	<ul style="list-style-type: none"> • Coeficiente de rugosidad de <i>Manning</i> para cauces

Tabla 1.1 Procesos, algoritmos y parámetros de *Hydrotel*

Como resultado se obtienen series diarias en cualquier segmento de la red de corrientes de las variables principales del modelo. Asimismo, se generan diversos mapas temáticos, gráficos y tablas disponibles a exportar.

1.4 Proyecto Physitel

Como ya se ha comentado, el modelo hidrológico se desarrolló en dos plataformas, *Physitel* e *Hydrotel*, donde primero se debe desarrollar el proyecto *Physitel*, que consta de 13 etapas a través de las cuales se definen las características de la cuenca, las cuales son exportadas al *Hydrotel* para la modelación hidrológica.



Etapa 1. Importación del modelo digital de elevación



Etapa 2. Determinación y edición de pendientes



Etapa 3. Importación y edición de la red vectorial de corrientes impuesta



Etapa 4. Determinación y edición de la red matricial de corrientes impuesta



Etapa 5. Determinación y edición del modelo digital de elevación modificado



Etapa 6. Determinación y edición de las direcciones de flujo



Etapa 7. Determinación del parteaguas de la cuenca



Etapa 8. Determinación de la red de corrientes modelada



Etapa 9. Reorientación de las celdas alrededor de los nodos

-  Etapa 10. Determinación y edición de las unidades hidrológicas
-  Etapa 11. Importación y edición de los usos de suelo
-  Etapa 12. Importación y edición de los tipos de suelo
-  Etapa 13. Exportación del proyecto *Physitel* al formato *Hydrotel*

1.4.1 Importación del modelo digital de elevación

Esta etapa consistió en incorporar a *Physitel* el modelo digital de elevaciones (Fig. 1.11), a partir del cual se definió la fisiografía de la cuenca.

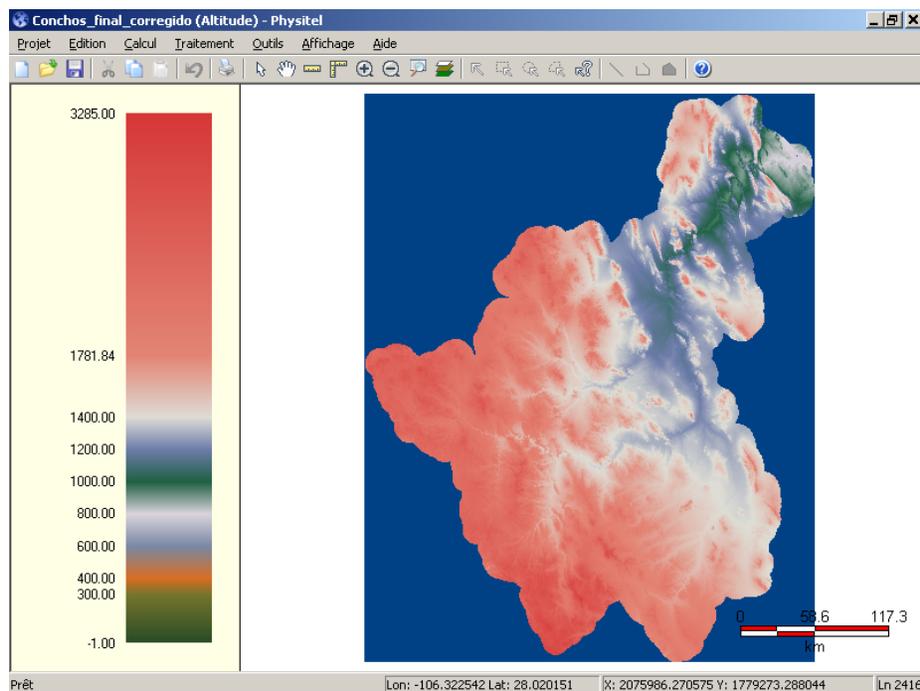


Figura 1.11 Modelo digital de elevaciones de la cuenca del río Conchos, etapa 1 de *Physitel*

1.4.2 Importación y edición de la red vectorial de corrientes impuesta

Esta etapa consiste en incorporar la red hidrográfica (lagos y ríos) bajo un formato vectorial (Fig. 3.3). Procesado en el apartado 1.1.6 (Fig. 1.12). Y a partir de la cual *Physitel* determinará una red matricial siguiendo la correcta ubicación de la red hidrológica vectorial impuesta.

La hidrografía fue utilizada para indicarle a la aplicación de *Physitel* la correcta trayectoria que sigue cada una de las corrientes (Ref. 2). Para ello se realizó una edición a la red mostrada en la figura. La edición consistió en quitar y conectar corrientes tributarias hasta obtener una red de drenaje compuesta solamente por corrientes principales, para ello se toma como base la red de drenaje que se genera en *Physitel* previamente. La red resultante es ingresada nuevamente al programa.

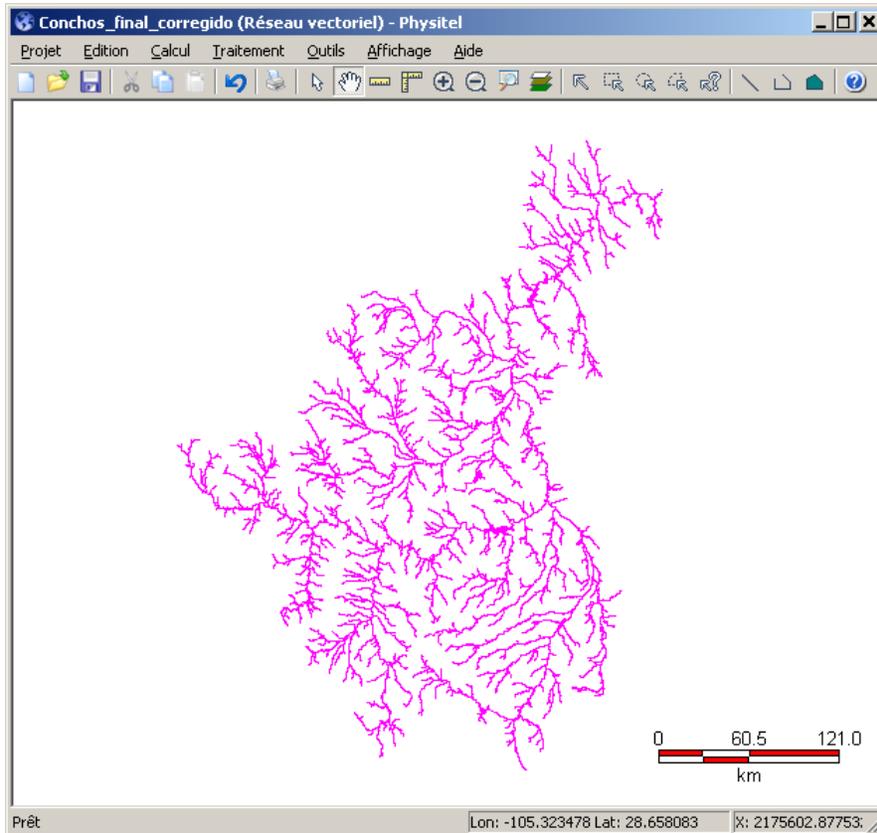


Figura 1.12 Red vectorial escala 1:50,000 dentro de *Physitel*

1.4.3 Determinación y edición de la red matricial de corrientes impuesta

La determinación y edición de la red matricial de corrientes (Fig. 1.13) consistió en obtener una representación matricial (Malla) de la red hidrográfica vectorial impuesta, es decir la red matricial resultante es en realidad el quemado de la red vectorial en el MDE a nivel celda.

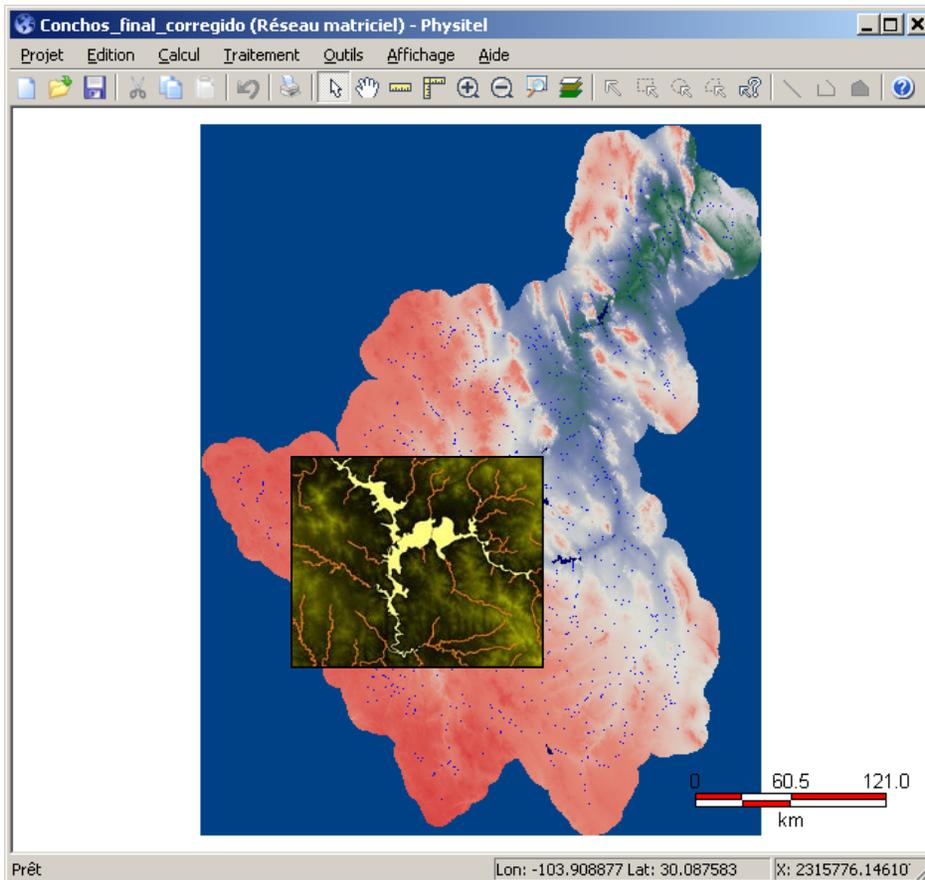


Figura 1.13 MDE y la red matricial generada a partir de la red vectorial 1:50,000 (INEGI)

1.4.4 Determinación y edición del modelo digital de elevación modificado

Esta etapa consistió en modificar el MDE por medio de un hundimiento de las elevaciones registradas sobre y en la periferia de la red matricial generada (Fig. 1.14), con el fin de dar una afluencia de los escurrimientos hacia las corrientes.

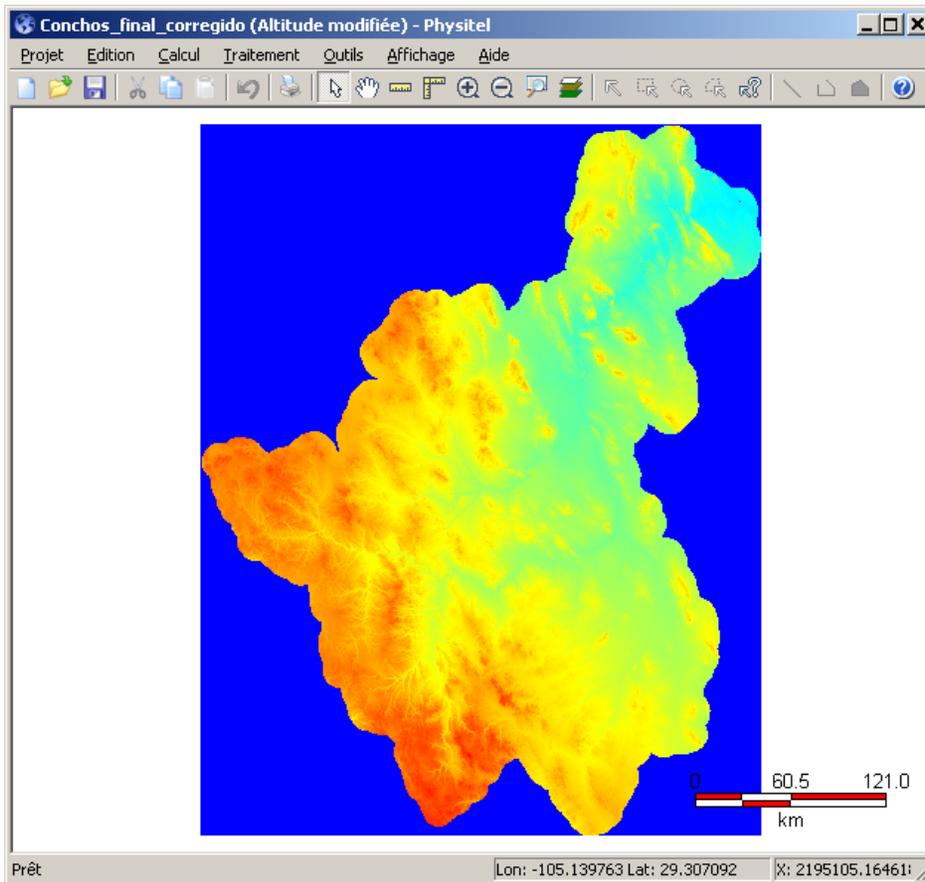


Figura 1.14 Muestra el área de influencia modificada en el MDE en función de las corrientes y cuerpos de agua

1.4.5 Determinación y edición de pendientes

Las pendientes son determinadas por *Physitel* en función de la elevación del terreno (MDE) y la distancia, es decir, el *Physitel* determina la pendiente para cada celda (Fig. 1.15) tomando en cuenta las elevaciones de las celdas vecinas.

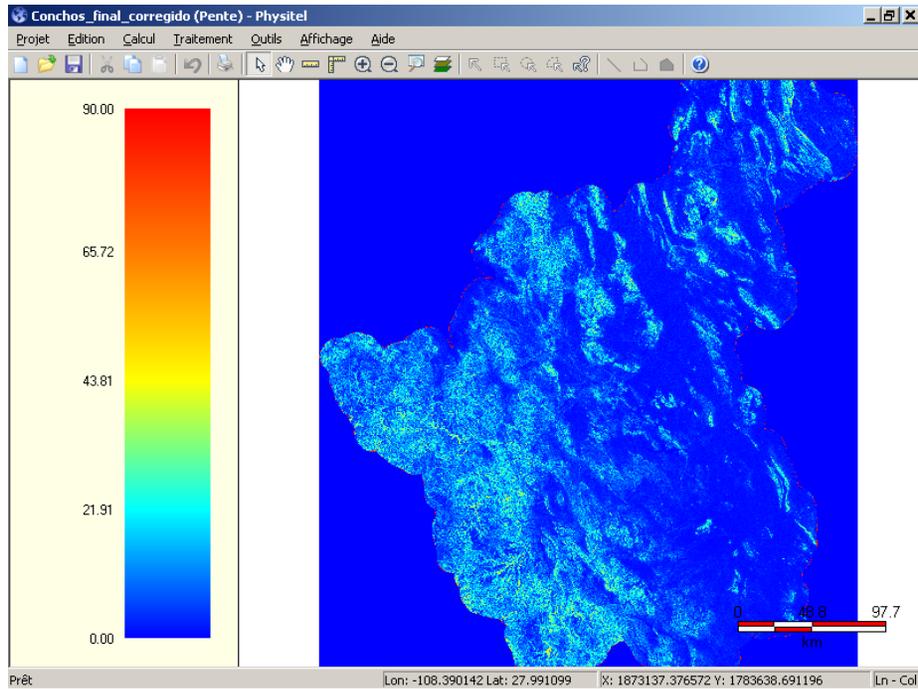


Figura 1.15 Pendientes generadas en la etapa 2 de Physitel a partir del MDE

1.4.6 Determinación y edición de las direcciones de flujo

En esta etapa *Physitel* le asigna una dirección de flujo a cada celda del MDE que constituye el área de estudio (Fig. 1.16). La dirección resultante para cada celda es producto del peso de la pendiente y así como la red matricial.

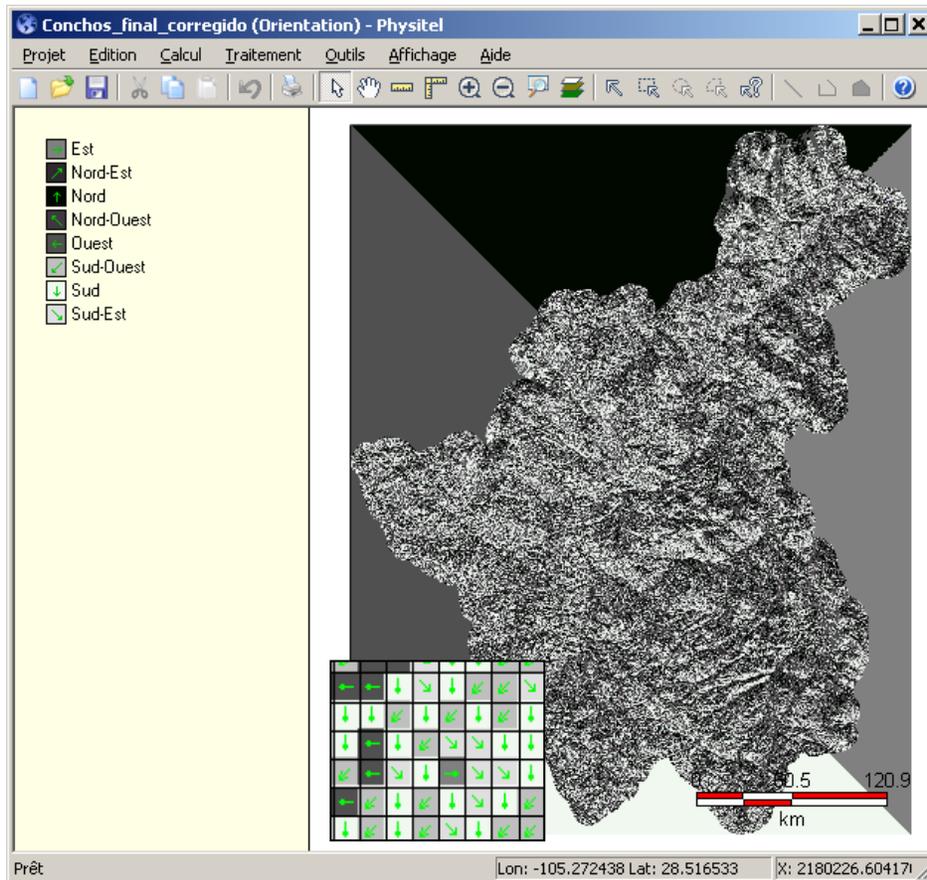


Figura 1.16 Muestra el resultado del cálculo realizado en la etapa 6, así como un acercamiento donde se puede observar las direcciones obtenidas a nivel celda

La edición de las direcciones consistió básicamente en asignar un sentido a las celdas indeterminadas que son celdas a las cuales *Physitel* no les dio una dirección de flujo.

1.4.7 Determinación del parteaguas de la cuenca

Con base en la conectividad de direcciones de flujo entre celdas *Physitel* determina el área de drenaje de la cuenca (parteaguas). Esta área está conformada por 8,314,054 celdas de 90 x 90 m equivalente a 67,343.84 km². Asimismo, se obtuvo la red de corriente total de la cuenca del Conchos (Fig. 1.18), esta red contiene ya todos los cuerpos de aguas seleccionados con anterioridad (apartado 3.1.6) para la modelación en *Hydrotel*.

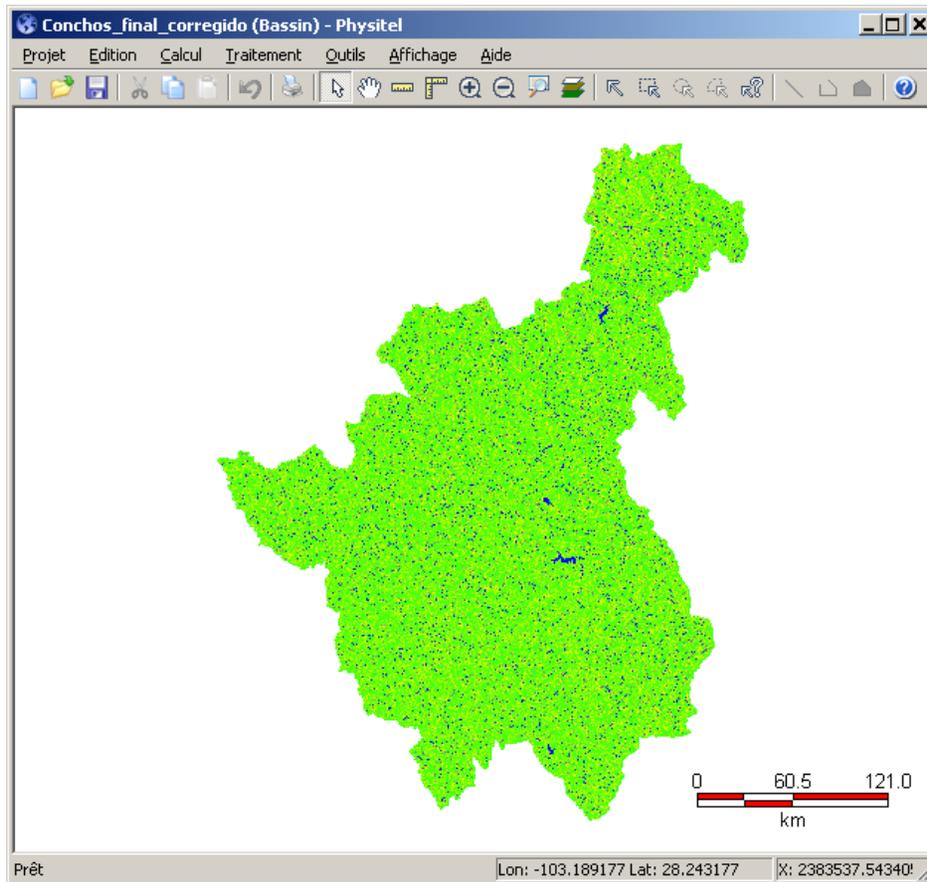


Figura 1.17 Parteaguas y red de corrientes resultado de la etapa 7 de *Physitel*

1.4.8 Determinación de la red de corrientes modelada

La etapa ocho de *Physitel* permitió obtener la red a modelar (Fig. 1.18), lo anterior se realizó usando un umbral de densidad de corrientes de 5,000.

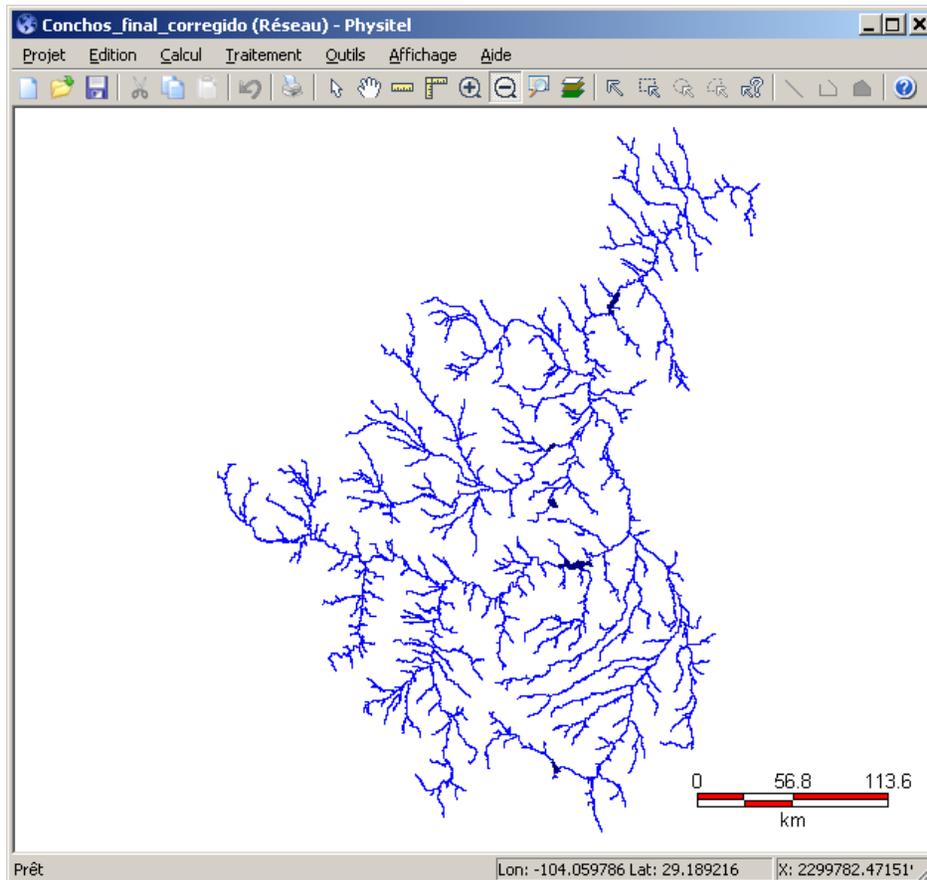


Figura 1.18 Muestra la red de corrientes a modelar

1.4.9 Determinación y edición de las unidades hidrológicas

Las unidades hidrológicas relativamente homogéneas (UHRH) son áreas de drenaje correspondientes a una corriente y a un nodo, éstas las relaciona y clasifica el sistema *Physitel* estableciendo una numeración en sentido horario de la corriente, comenzando la enumeración sobre la margen izquierda de las corrientes. Para la cuenca del Conchos se obtuvieron 954 unidades hidrológicas relativamente homogéneas (Fig. 1.19), que en promedio tienen 67.08 km².

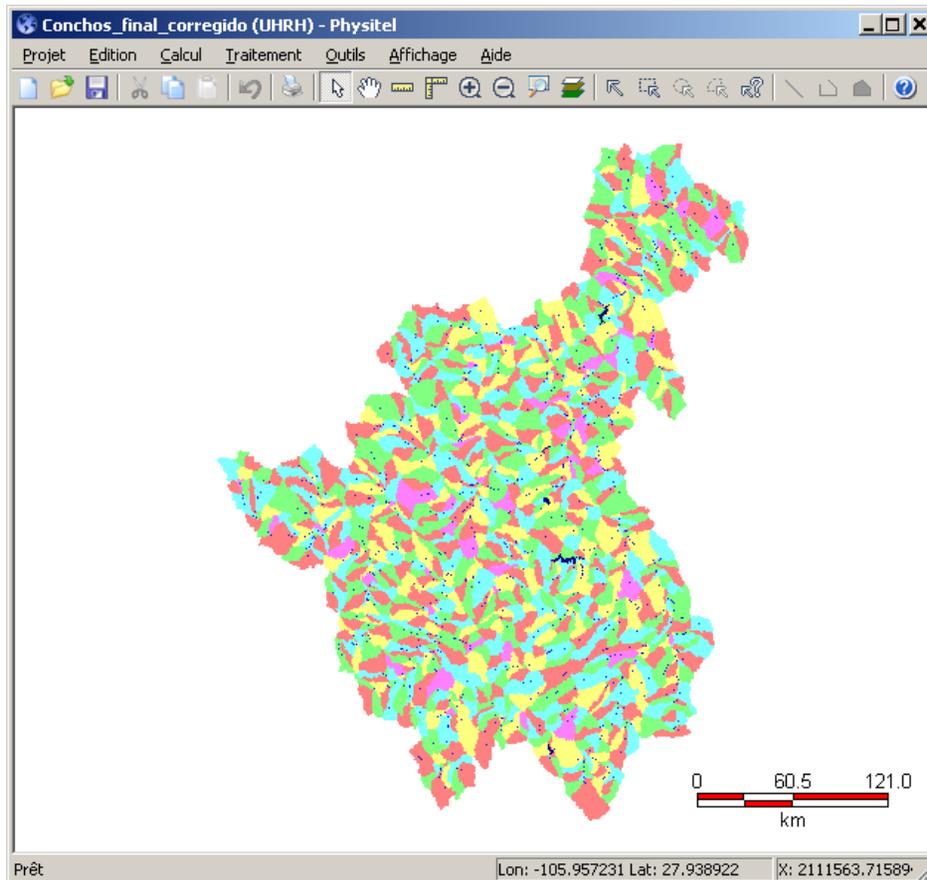


Figura 1.19 Muestra las UHRH obtenidas para la cuenca del Conchos

1.4.10 Importación y edición de los usos de suelo

La capa vectorial de uso de suelo, se incorpora al modelo a través de *Physitel* en formato *.txt* obtenida por un proceso de transformación a formato matricial. El resultado es una malla en la que cada celda contiene una sola clase de uso de suelo. En total *Physitel* reconoce 19 clases de uso de suelo, estas se transforman a porcentajes de uso por UHRH (Fig. 1.20).

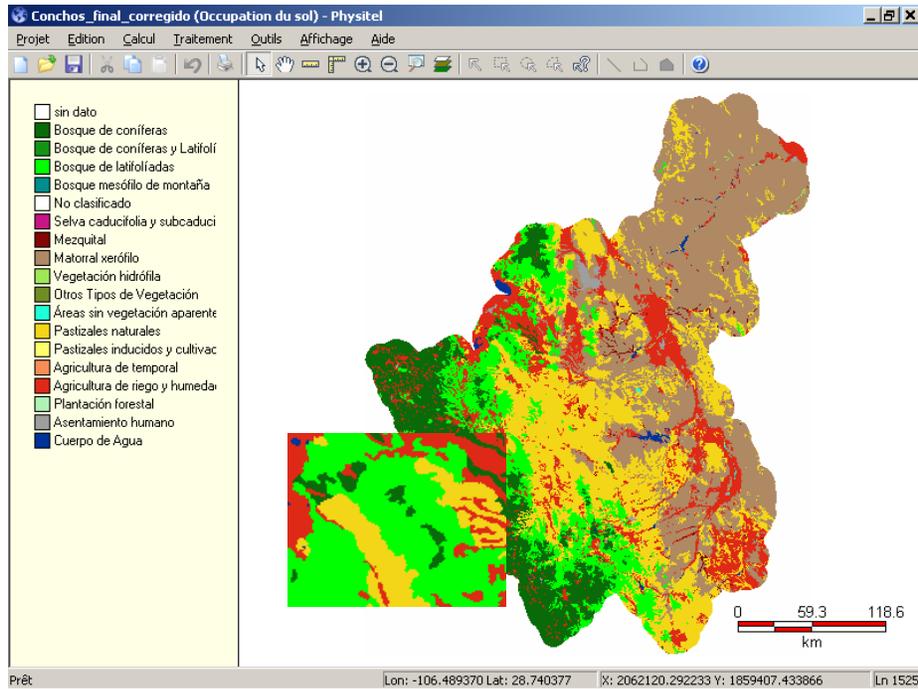


Figura 1.20 Muestra los diferentes usos de suelo en la cuenca del río Conchos

1.4.11 Importación y edición de los tipos de suelo

Al igual que para el uso de suelo, la textura del suelo es importada a *Physitel* a través del formato .txt, producto de la conversión a formato matricial de las clases texturales de la edafología. Finalmente, cada celda del mapa matricial contiene una textura representativa del suelo considerando el rodal dominante. En esta etapa *Physitel* calcula en porcentajes el tipo de suelo para cada UHRH (Fig. 1.21).

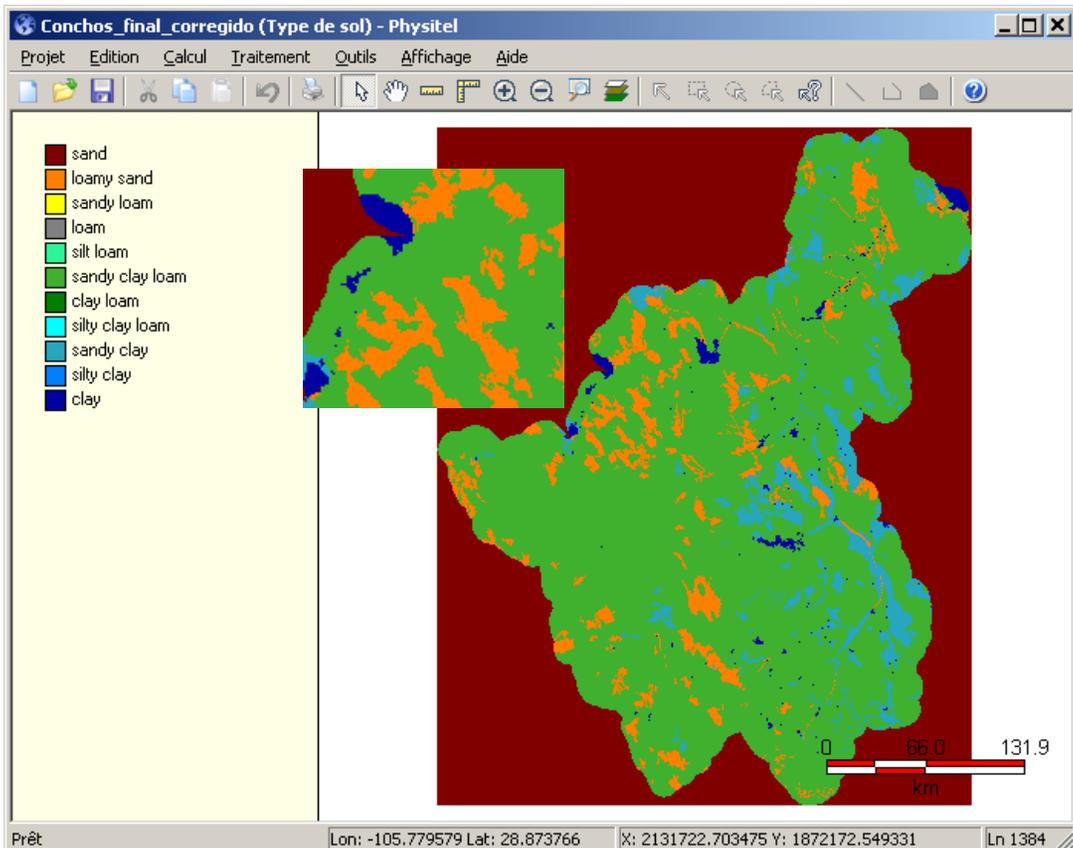


Figura 1.21 Muestra la edafología de la cuenca del río Conchos en el ambiente Physitel

1.4.12 Exportación del proyecto *Physitel* al formato *Hydrotel*

La exportación es la última etapa del *Physitel*, consiste en cambiar la información (Fig. 1.22) al formato *Hydrotel*. Los archivos transformados son exportados a una carpeta llamada *Hydrotel*.

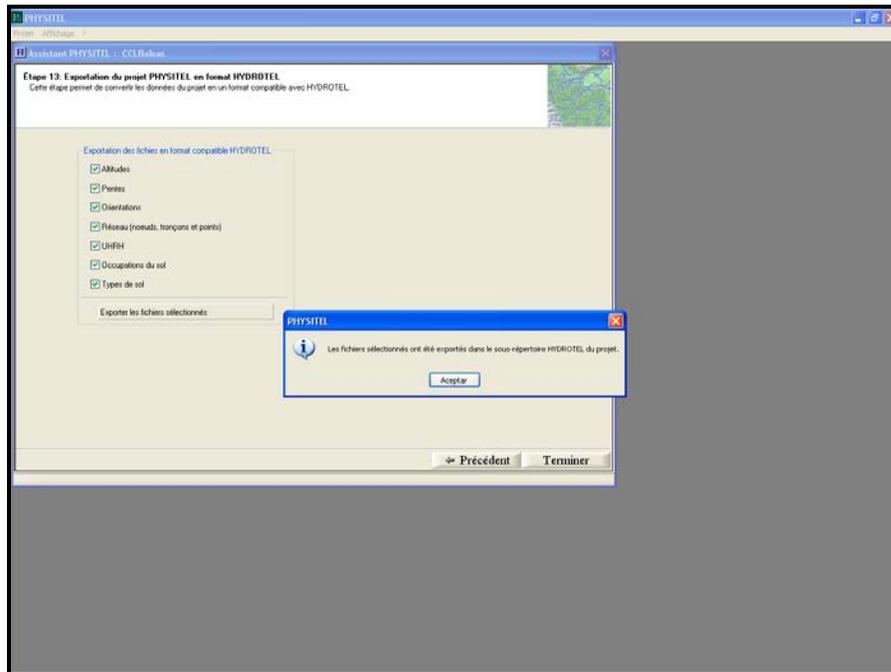


Figura 1.22 Muestra las coberturas exportadas a formatos hydrotel

1.5 Proyecto Hydrotel

En la creación del proyecto *Hydrotel* es necesario importar los archivos procesados por *Physitel*, es decir, la fisiografía de la cuenca. Así también, archivos que contienen información meteorológica, hidrométrica e información de vegetación. Todo esto con el fin de representar de una forma más aproximada las condiciones naturales de la cuenca y aprovechar al máximo la información disponible.

1.5.1 Importación de archivos Physitel

Los archivos que se exportaron de formato *Physitel* a formato *Hydrotel* son: UHRH, altitudes, orientaciones, pendientes, nodos, tramos, puntos, edafología, tipo de suelo y propiedades hidráulicas. En la figuras 1.23 y 1.24 se muestra la interfaz en donde se indica la ubicación de los archivos exportados por *Physitel*.

Nouveau projet - Étape 2 de 4

Structure terrestre

UHRH ...

Altitudes ...

Orientations ...

Pentes ...

Réseau hydrographique

Noeuds ...

Tronçons ...

Points ...

>

Figura 1.23 Interfaz de importación de archivos de fisiografía de la cuenca para el montaje del proyecto en Hydrotel

Nouveau projet - Étape 3 de 4

Occupation du sol

Distribution spatiale ...

Types de sol

Distribution spatiale ...

Propriétés hydrauliques ...

>

Figura 1.24 Interfaz de importación de archivos de uso y tipo de suelo de la cuenca para el montaje del proyecto en Hydrotel

Durante el proceso de importación de los archivos *Physitel* se crean carpetas en el subdirectorio donde se colocó el proyecto, las carpetas que se crean son *Physitel*, *Physio*, *Hydro*, *Meteo* y *HGM*, en donde estas últimas cuatro carpetas se encuentran vacías. A las carpetas *Physio*, *Hydro* y *Meteo* es necesario producirles sus archivos de alimentación al proyecto manualmente.

1.5.2 Definición de archivos fisiográficos

En la carpeta *Physio* es necesario agregar dos archivos en formato .txt, esto para que el modelo hidrológico cuente con la información del índice foliar y profundidad de raíces distribuidas en toda la cuenca. El índice foliar se encuentra contenido en el archivo *LAI.txt* (Fig. 1.25) y la profundidad de raíces en el archivo *Raices.txt* (Fig. 1.26).

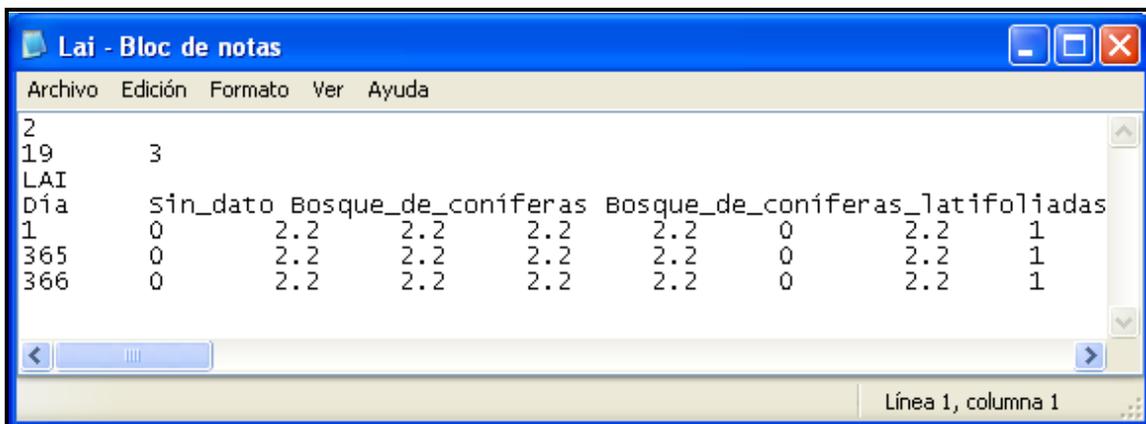


Figura 1.25 Archivo *Lai.txt* (índice foliar)

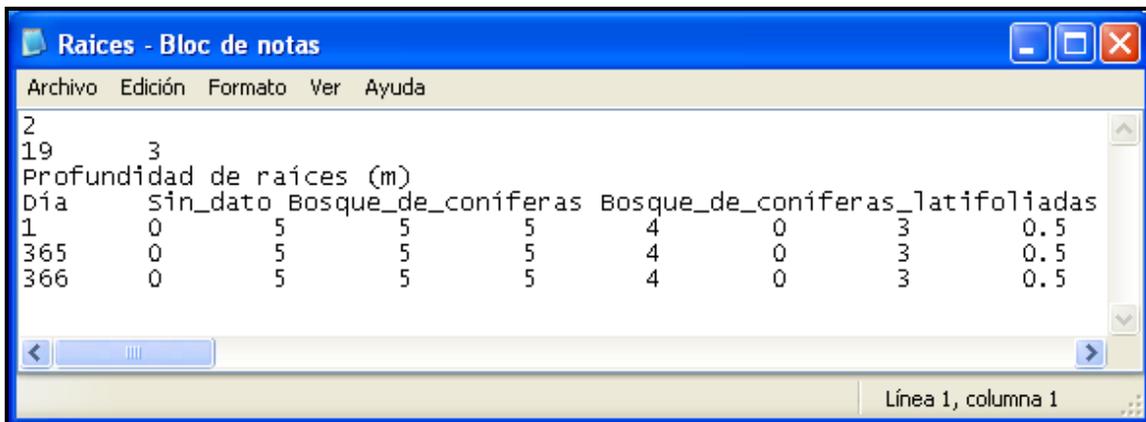


Figura 1.26 Archivo *Raices.txt* (profundidad de raíces)

1.5.3 Importación de archivos hidrométricos y meteorológicos

Para alimentar al modelo hidrológico es necesario crear archivos de control que contienen varios parámetros, de los cuales los más importantes son claves, coordenadas y altitud en el caso de

las estaciones meteorológicas (Fig. 1.27 y 1.28). Estos archivos se colocan en la carpeta *Meteo* e *Hydro* respectivamente junto con sus series históricas.

Se prepararon las series históricas completas de temperatura mínima, temperatura máxima y precipitación diarias en formato legible por *Hydrotel*. Asimismo, se prepararon las series históricas completas de escurrimiento diario en tres estaciones de control ubicadas en o cerca de la salida de las subcuencas.

meteo1.stm - WordPad

Archivo Edición Ver Insertar Formato Ayuda

Courier New 10 Occidental

125

Estaciones climatológicas

08112MH	2086031.444	1852332.08	1699	3	@6	meteo\
08113MH	2093064.194	1839547.817	1574	3	@6	meteo\
08114MH	2085109.648	1846854.213	1789	3	@6	meteo\
08120MH	2173506.251	1798406.463	1201	3	@6	meteo\
08123MH	2038477.261	1765321.735	1670	3	@6	meteo\
08126MH	1950488.603	1778068.93	2403	3	@6	meteo\
08127MH	2061310.593	1794640.387	1610	3	@6	meteo\
08130MH	2122010.651	1643134.303	1949	3	@6	meteo\
08133MH	2214379.858	1720128.872	1314	3	@6	meteo\
08135MH	2176799.509	1778268.29	1189	3	@6	meteo\
08136MH	2096974.768	1771358.658	1380	3	@6	meteo\
08137MH	1972903.873	1763937.418	2461	3	@6	meteo\
08147MH	2110327.425	1854125.522	1346	3	@6	meteo\
08151MH	2164100.344	1657503.776	1597	3	@6	meteo\
08152MH	2124516.272	1715663.109	1340	3	@6	meteo\
08153MH	2124891.212	1715346.194	1352	3	@6	meteo\
08156MH	2186883.73	1634778.553	1515	3	@6	meteo\
08157MH	2118803.393	1868625.649	1275	3	@6	meteo\
08158MH	2125488.638	1773796.335	1316	3	@6	meteo\
08162MH	2186928.384	1740971.9	1215	3	@6	meteo\
08165MH	2101375.371	1846381.526	1415	3	@6	meteo\
08169MH	2078423.935	1672278.244	1566	3	@6	meteo\
08174MH	2160270.813	1791346.024	1200	3	@6	meteo\

Para obtener Ayuda, presione F1

Figura 1.27 Archivos de control de las estaciones seleccionadas meteorológicas

hydro.sth - WordPad

Archivo Edición Ver Insertar Formato Ayuda

Courier New 10 Occidental

2
23
commentaire

0024181	2129311.818930	1774002.276540	0	3	05	hydro
0024182	2186785.002390	1740668.050580	0	3	05	hydro
0024195	2167273.656070	1601527.908030	0	3	05	hydro
0024213	2168587.920730	1728036.045060	0	3	05	hydro
0024215	2168587.920730	1728036.045060	0	3	05	hydro
0024216	2165230.843230	1724891.975240	0	3	05	hydro
0024218	2144815.793400	1794032.632500	0	3	05	hydro
0024221	2183117.743460	1765930.428710	0	3	05	hydro
0024225	2211759.174270	1678755.973600	0	3	05	hydro
0024226	2165670.148370	1835013.007150	0	3	05	hydro
0024234	2147530.007490	1793687.738960	0	3	05	hydro
0024242	2186153.357130	1742984.832690	0	3	05	hydro
0024258	2139289.799690	1608187.547580	0	3	05	hydro
0024259	2146685.843340	1792602.798530	0	3	05	hydro
0024280	2139289.799690	1608187.547580	0	3	05	hydro
0024282	2139289.799690	1608187.547580	0	3	05	hydro
0024313	2099696.055120	1844584.304690	0	3	05	hydro
0024331	2099722.414480	1845506.113670	0	3	05	hydro
0024339	2179334.943880	1880364.468140	0	3	05	hydro
0024346	2130871.783340	1655783.512200	0	3	05	hydro
0024353	2184735.520900	1691604.470900	0	3	05	hydro
0024388	2236814.281210	1939613.559160	0	3	05	hydro
0024400	2081698.085770	1631382.784880	0	3	05	hydro

Para obtener Ayuda, presione F1

Figura 1.28 Archivos de control de las estaciones seleccionadas hidrométricas

Las series históricas meteorológicas contienen información diaria de precipitación, temperatura máxima y temperatura mínima, de las cuales el modelo hidrológico genera un escurrimiento sobre la cuenca. Las series históricas hidrométricas contienen información diaria de caudales que sirven de comparación para realizar la calibración del modelo. Todo esto para cada una de las estaciones meteorológicas e hidrométricas contenidas en sus respectivos archivos de control. En la figura 1.29 y 1.30 se muestra la ubicación de las estaciones meteorológicas e hidrométricas respectivamente en plataforma *Hydrotel*.

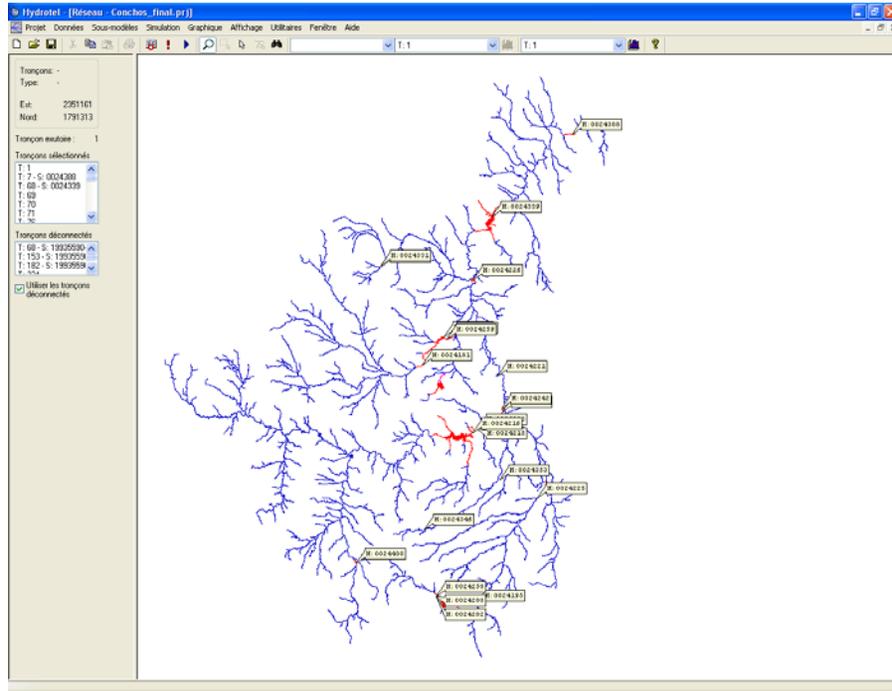


Figura 1.29 Localización de estaciones hidrométricas

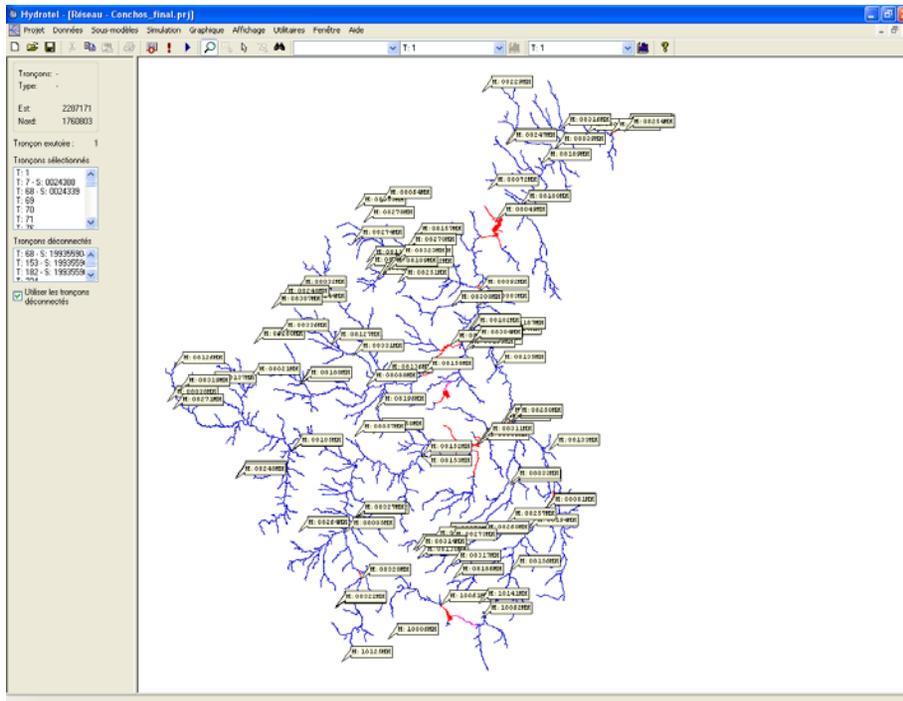


Figura 1.30 Localización de estaciones meteorológicas

1.5.4 Creación de grupos de UHRH

Para calibrar el modelo hidrológico es necesario crear grupos, los cuales deben contar con una estación de calibración situada a la salida de cada grupo (subcuenca), para así en cada uno de ellos modificar algunos parámetros fisiográficos e hidráulicos. Se integraron dos grupos formados dentro del modelo hidrológico (Fig. 1.31).

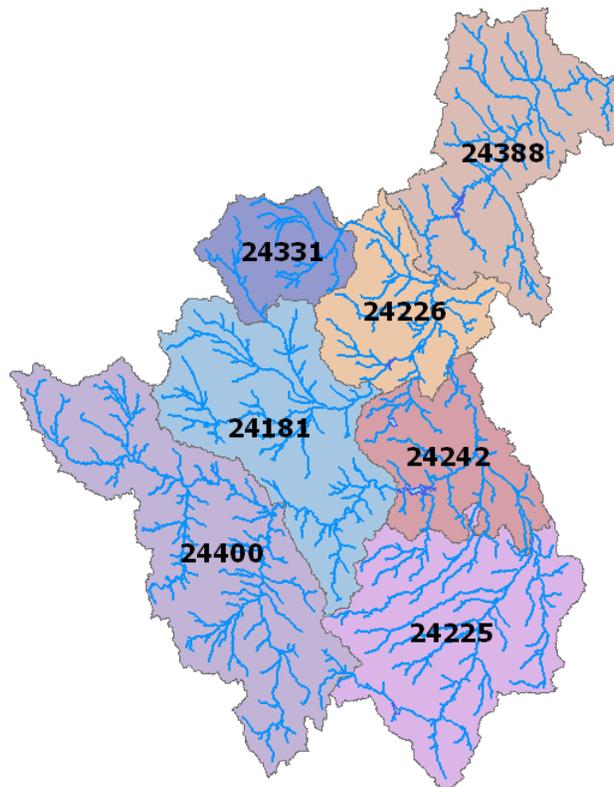


Tabla 1.31 Grupos formados en la cuenca

1.5.5 Definición de gradientes temperatura

Una de las funciones de los modelos de simulación de *Hydrotel* es el de interpolar los datos medidos en las tres estaciones meteorológicas más cercanas a la UHRH (precipitación, temperatura máxima y temperatura mínima), para obtener una distribución espacial de esas variables.

Cuando se interpolan las temperaturas del aire en cada estación meteorológica sobre las UHRH se toma en cuenta la variación espacial de los valores, pero también es importante definir la variación espacial con respecto a la altitud entre la estación y cada UHRH. *Hydrotel* corrige las temperaturas por cada 100 m de altura en cada grupo, la ecuación de corrección es la siguiente:

$$T_c = T_m + G_T \left(\frac{A_s - A_u}{100} \right)$$

Donde:

- T_c = temperatura corregida (°C)
- T_m = temperatura medida (°C)
- G_T = gradiente vertical de la temperatura (°C / 100 m)
- A_s = altitud de la estación (m)
- A_u = altitud de la UHRH (m)

Para obtener los gradientes de temperatura de cada grupo (Tabla 3.1) primero se determinaron las temperaturas promedio anuales de cada estación, estas se agruparon conforme a los grupos de calibración para así calcular el gradiente vertical de temperatura. En la interfaz de aplicación de *Hydrotel* se introducen cada uno de los valores calculados (Fig. 1.32).

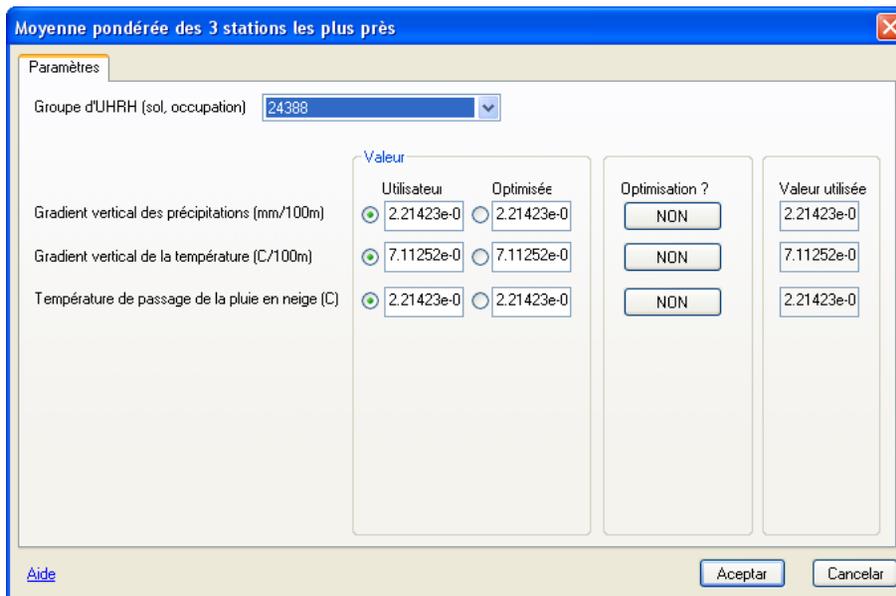


Figura 1.32 Interfaz de gradientes verticales de temperatura

En la tabla 3.2 se muestran los gradientes de temperatura que se obtuvieron para cada grupo.

Grupos	24388	24331	24226	24242	24225	24181	24400
Gradiente de temperatura	-0.6149	-0.4715	-0.6272	-0.0018	-0.4811	-0.7687	-0.5909

Tabla 1.2 Valores de gradientes verticales de temperatura

1.5.6 Generación de hidrogramas geomorfológicos

Una vez creado el proyecto *Hydrotel* para la modelación hidrológica de la cuenca se procede a simular el modelo en toda el área de proyecto. Esta simulación se realiza para crear los archivos de los hidrogramas geomorfológicos (HGM) dentro de la carpeta *HGM*.

Los archivos *HGM* contienen los hidrogramas geomorfológicos de cada UHRH (Fig. 1.33), es decir, hidrogramas que describen la capacidad de respuesta del escurrimiento en cada UHRH. Los HGM dependen del coeficiente de *Manning* y de la pendiente de los UHRH.

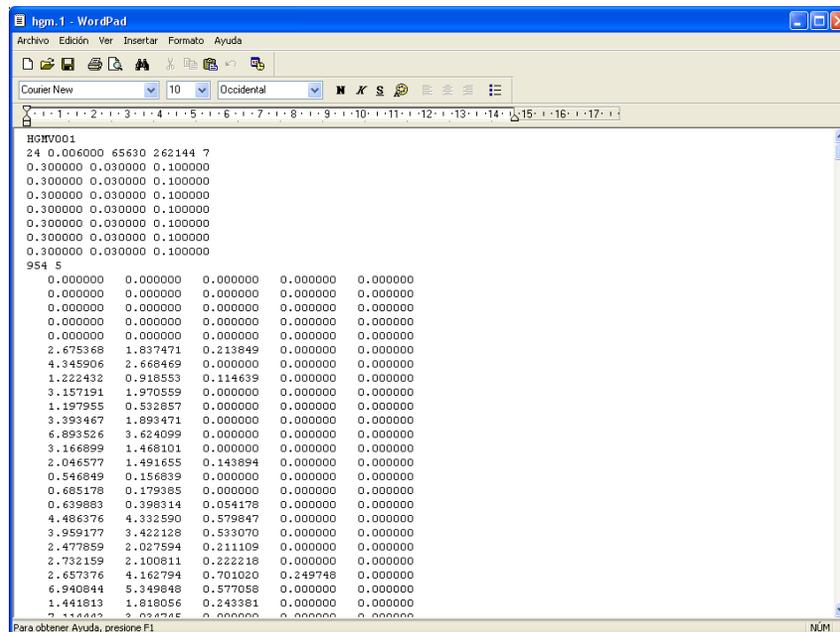


Figura 1.33 Archivo de datos del hidrograma geomorfológico

Los archivos *HGM* se generan una sola vez al iniciar la primera simulación para toda la etapa de proyecto al menos que haya un cambio en la conformación de los grupos o coeficiente de *Manning*.

1.5.7 Calibración del modelo hidrológico

Una vez concluida la etapa de generación de archivos *HGM*, como se mencionó en el inciso anterior, se procede a la calibración de grupos siguiendo el procedimiento que se enumera a continuación:

1. Elección del tramo de la red hidrográfica que funcionará como salida de la subcuenca, es decir, del grupo que será calibrado.
2. Elección de submodelos. Estos son los algoritmos alternativos vistos en el inciso 2.2.

3. Simulación de calibración. La simulación se realiza con un periodo de tiempo de X años y como condiciones iniciales se ajustan las humedades de las capas 1, 2 y 3.

En esta etapa los parámetros de ajuste son:

- *Humedades de las capas 1, 2 y 3 para las condiciones iniciales*
- *Factor de evapotranspiración*
- *Profundidades de las capas 1, 2 y 3*
- *Coefficiente de recesión*

La calibración de cada grupo se realiza con procedimientos manuales iterativos, apoyados con un esquema de optimización de parámetros, considerando el ajuste en volumen, forma y sincronización del hidrograma.

El ajuste en volumen se realizó obteniendo el mínimo porcentaje del error relativo de la diferencia entre el volumen observado (dado por los registros de las series hidrométricas) y el volumen simulado, tal y como se muestra a continuación:

$$Error\% = \frac{(Vol\ sim - Vol\ obs)}{Vol\ obs}$$

La forma y sincronización del hidrograma se llevó a cabo considerando el mayor valor de un índice estadístico, este índice es el coeficiente llamado *Nash Sutcliffe* que considera el caudal simulado, el caudal observado y el caudal medio en el periodo simulado, la ecuación utilizada está dada de la siguiente manera:

$$Nash = 1 - \frac{\sum_{i=1}^n (Qs_i - Qo_i)}{\sum_{i=1}^n (\bar{Q} - Qo_i)}$$

Donde:

- Nash* = Índice de error estadístico
Qs_i = Caudal simulado al paso de tiempo *i*
Qo_i = Caudal observado al paso de tiempo *i*
 \bar{Q} = Caudal medio en el periodo de la simulación considerada
i = Índice de paso de tiempo
n = Número de paso de tiempo

En la tabla 1.3 se muestran los valores de calibración de los grupos de la cuenca, el índice de error estadístico *Nash Sutcliffe* y el error relativo de la diferencia entre volúmenes.

Grupo	24388	24331	24226	24242	24225	24181	24400
Periodo	70-82	73-82	70-82	70-82	75-85	70-82	75-85
Factor ETP	1	1.8	0.6	1.8	2	1.8	0.95
Espesor ₁	0.1	0.1	0.1	0.05	0.1	0.1	0.08
Espesor ₂	0.5	0.34	0.5	0.2	0.23	0.34	0.12
Espesor ₃	1.5	0.35	1.5	1	0.7	0.35	2.1
CR	1.00E-7	1.00E-10	1.00E-7	1.00E-10	1.00E-11	1.00E-10	1.00E-11
$\Delta V(\%)$	-88.9	310.9	-59.4	-87.5	0.5	14.7	15.4
NS	-0.09	-32.11	0.35	-0.03	0.73	-1.46	-0.69

Tabla 1.3 Valores de calibración de grupos

En las figuras 1.34a y 1.35a, se muestran las subcuencas que se utilizaron para la calibración de los grupos 1 y 24331 respectivamente, y en las figuras 1.34b y 1.35b, la comparación de los hidrogramas observados con los hidrogramas simulados.

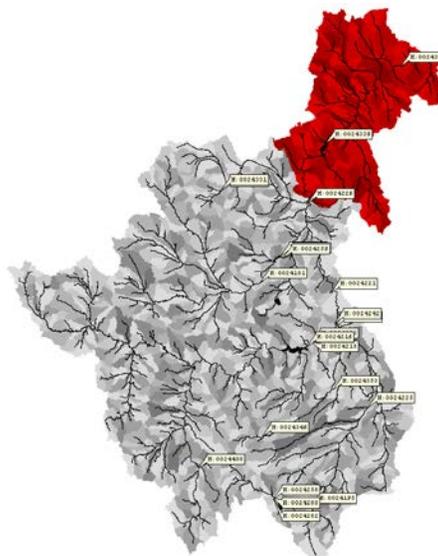


Figura 1.34a Calibración del grupo 1, subcuenca 24388

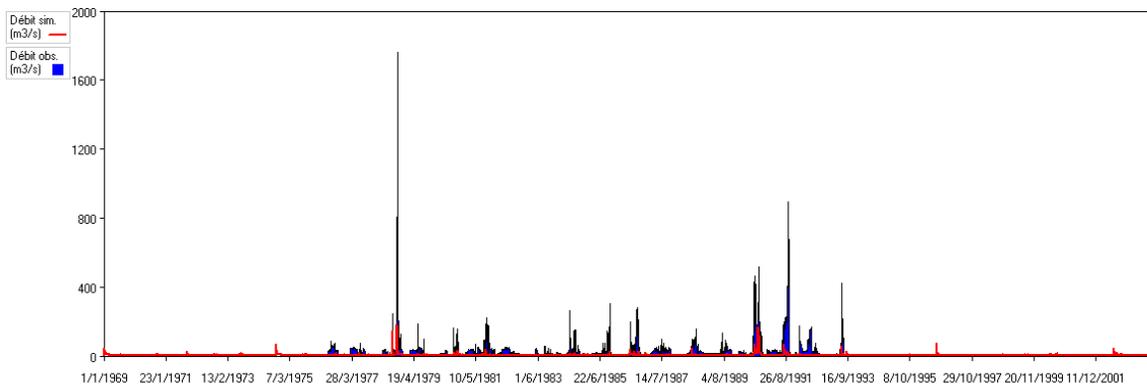


Figura 1.34b Hidrograma de calibración del grupo 24388

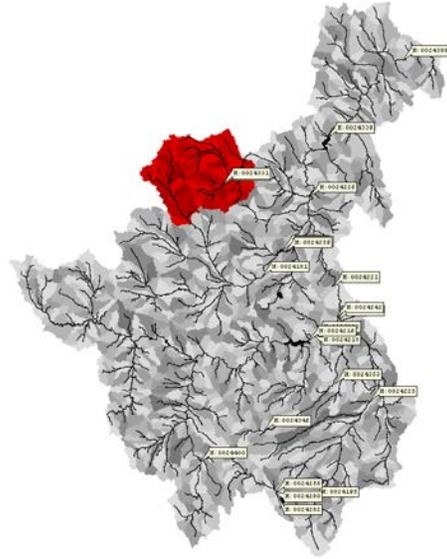


Figura 1.35a Calibración del grupo 24331

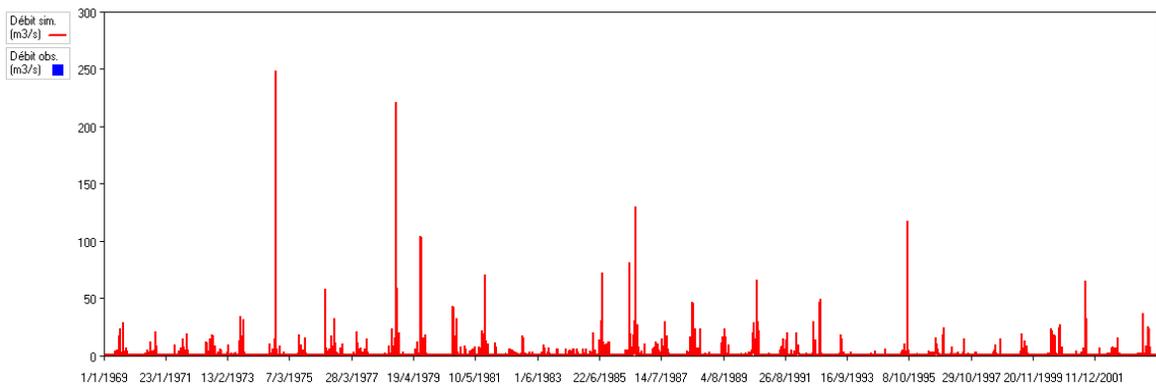


Figura 1.35b Hidrograma de calibración del grupo 2, subcuena Conchos

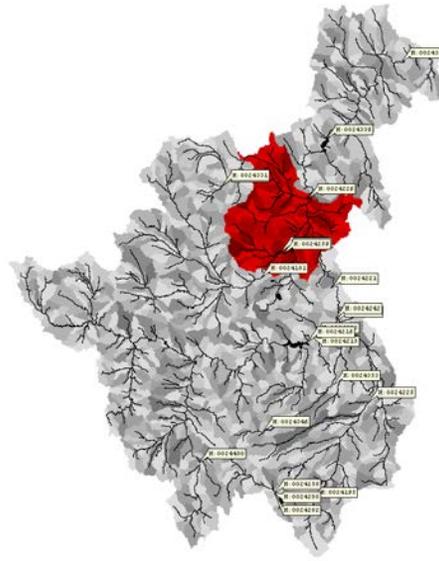


Figura 1.36a Calibración del grupo 24226

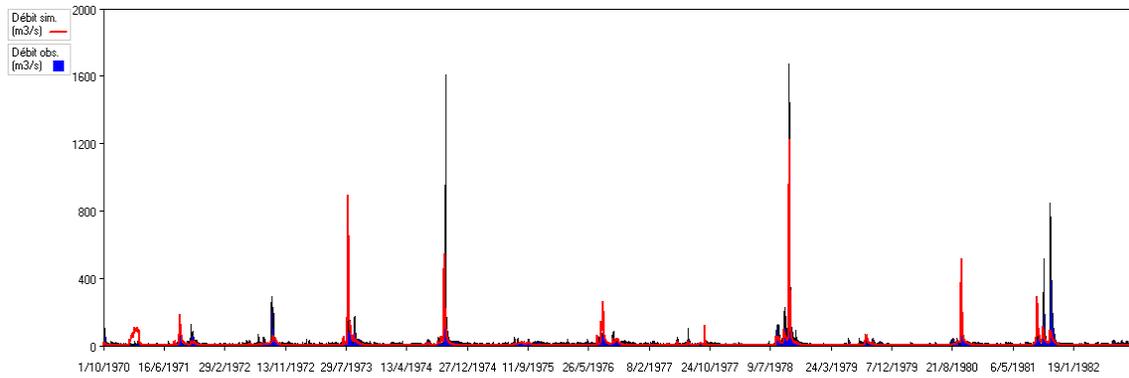


Figura 1.36b Hidrograma de calibración del grupo 24226

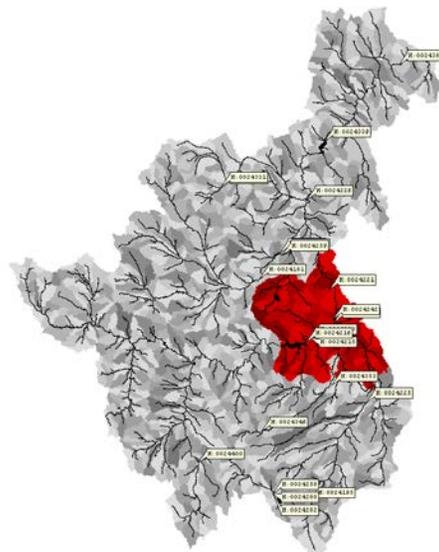


Figura 1.37a Calibración del grupo 24242

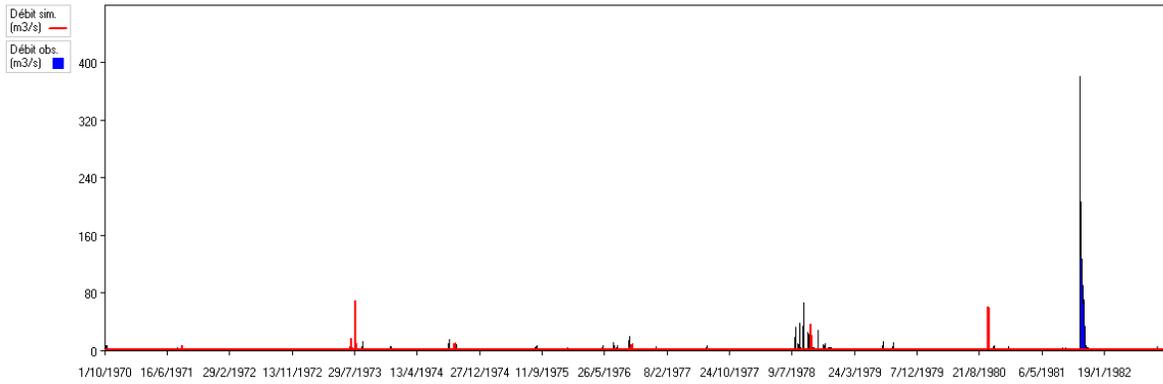


Figura 1.37b Hidrograma de calibración del grupo 24242

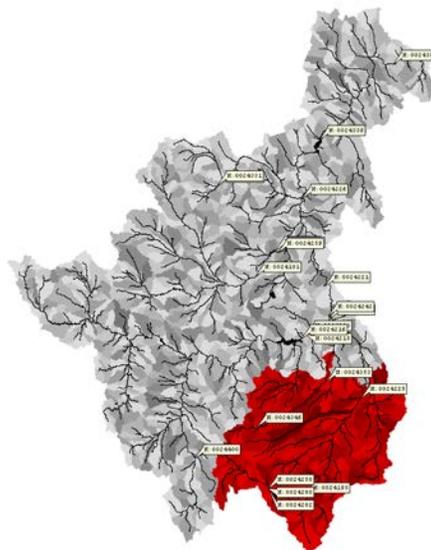


Figura 1.38a Calibración del grupo 24225

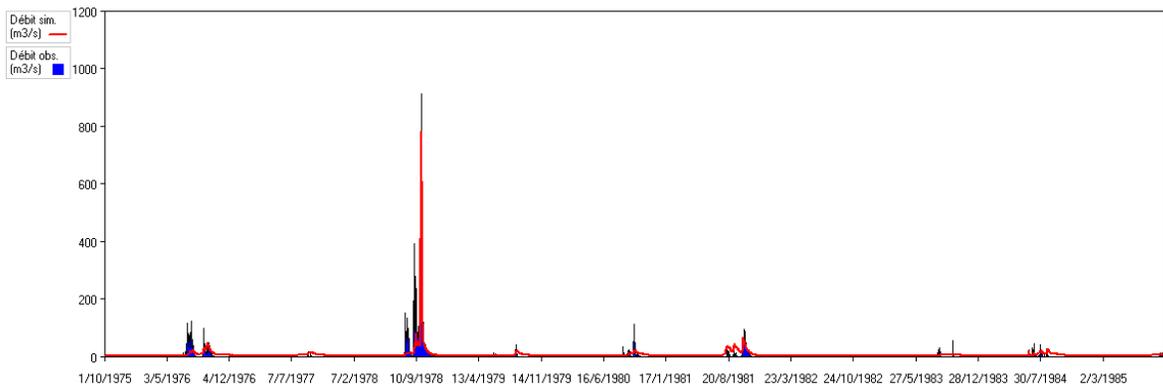


Figura 1.38b Hidrograma de calibración del grupo 24225

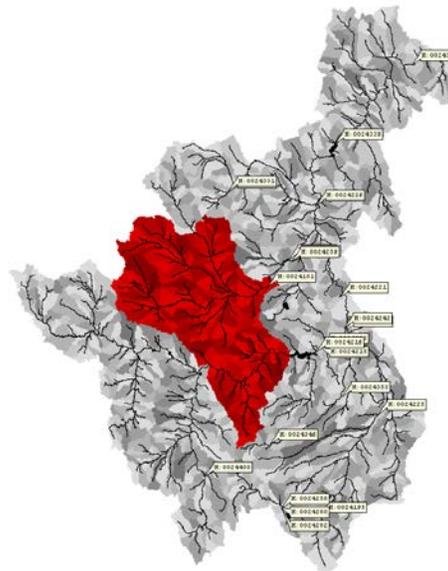


Figura 1.39a Calibración del grupo 24181

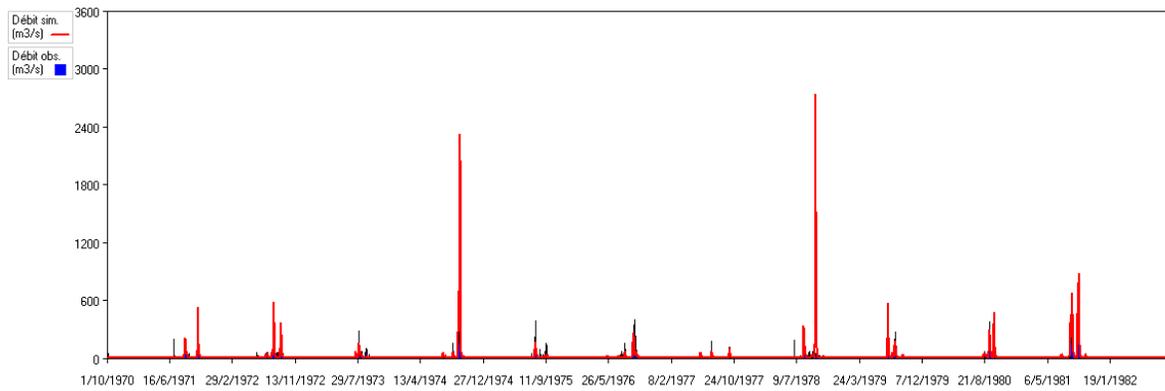


Figura 1.39b Hidrograma de calibración del grupo 24181

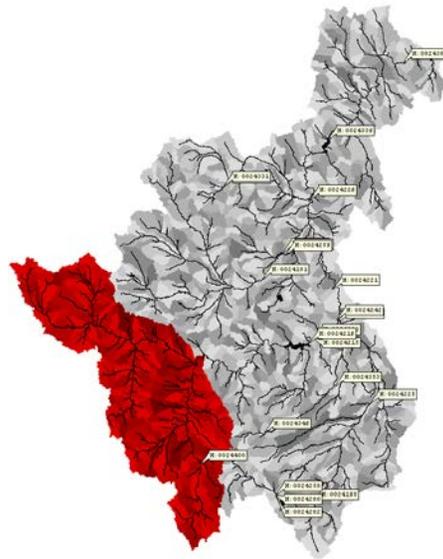


Figura 1.40a Calibración del grupo 24400

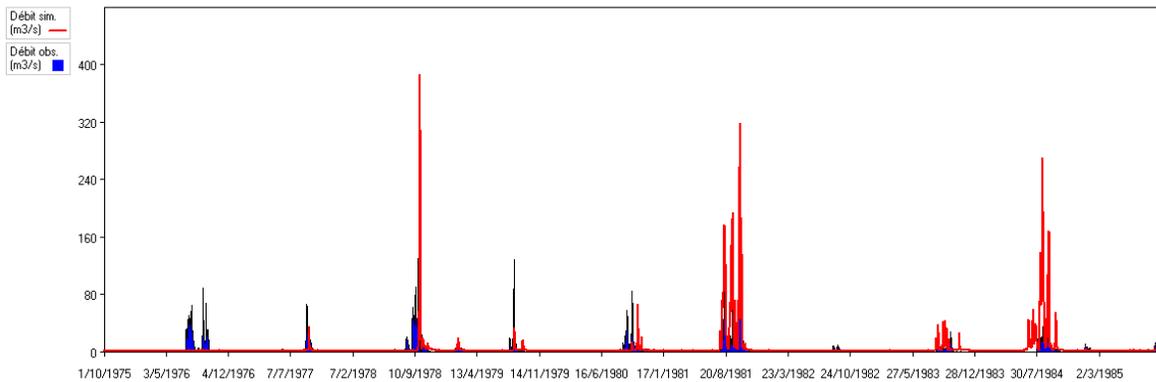


Figura 1.40b Hidrograma de calibración del grupo 24400

1.5.8 Simulación histórica del modelo hidrológico

Una vez obtenidos los parámetros de calibración se simuló una serie histórica de 35 años del 01 de enero de 1969 al 30 de diciembre del 2003. Sólo en este periodo se cuenta con información meteorológica. Los escurrimientos resultantes se obtuvieron por cuenca propia para las subcuencas de la figura 1.42 y sus hidrogramas se muestran de la figura 1.43 a la 1.49.

Figura 1.41 Subcuencas de simulación histórica

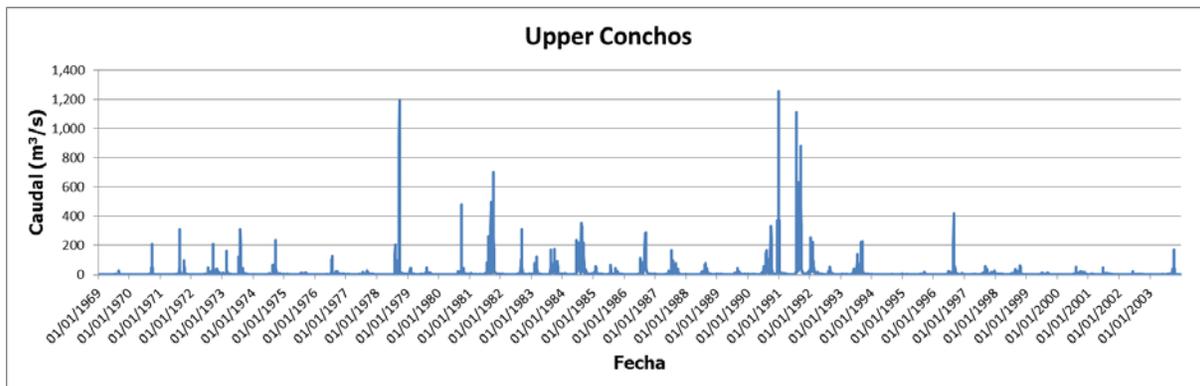


Figura 1.42 Subcuencas de simulación histórica

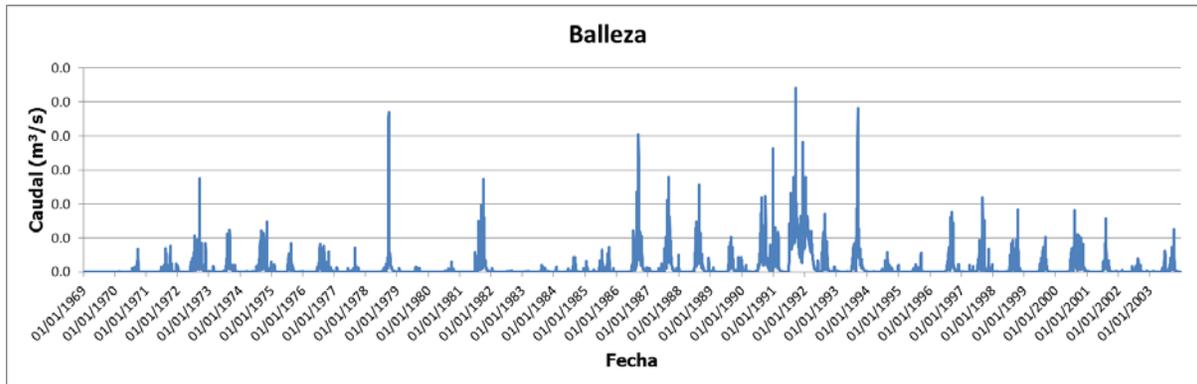


Figura 1.43 Subcuencas de simulación histórica

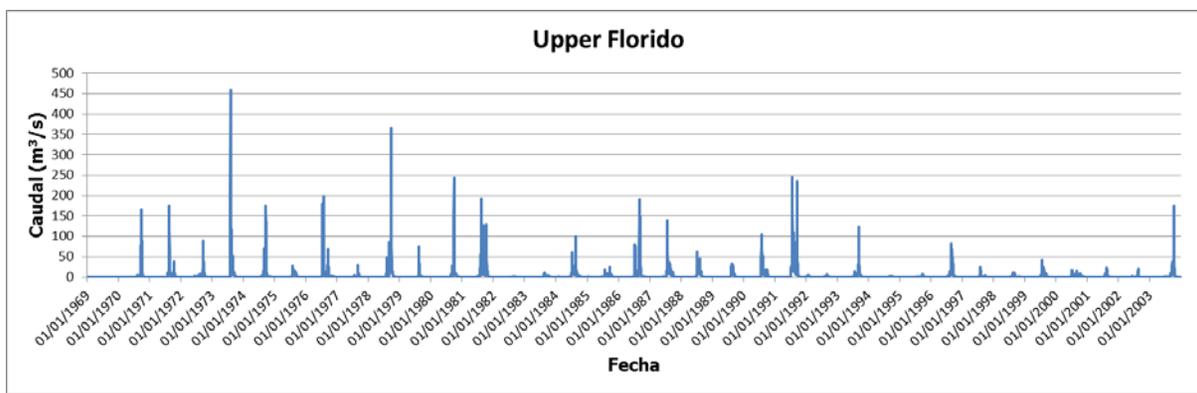


Figura 1.44 Subcuencas de simulación histórica

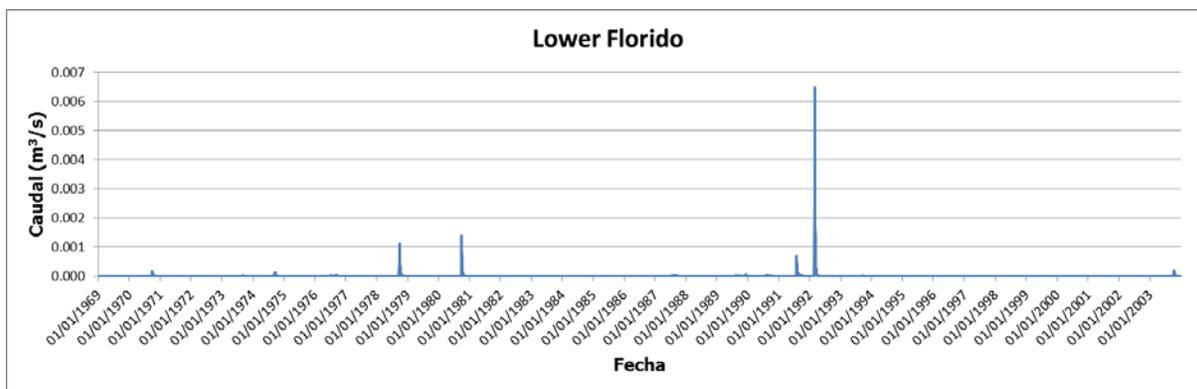


Figura 1.45 Subcuencas de simulación histórica

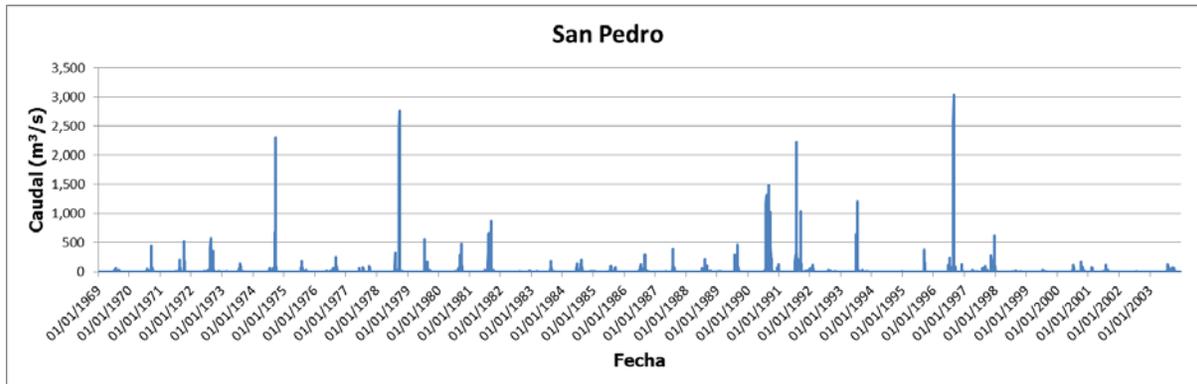


Figura 1.46 Subcuencas de simulación histórica

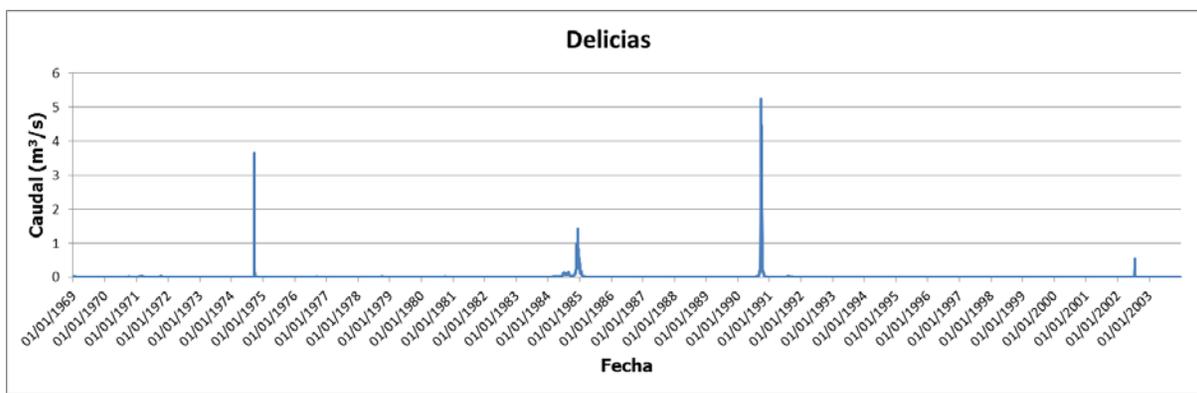


Figura 1.47 Subcuencas de simulación histórica

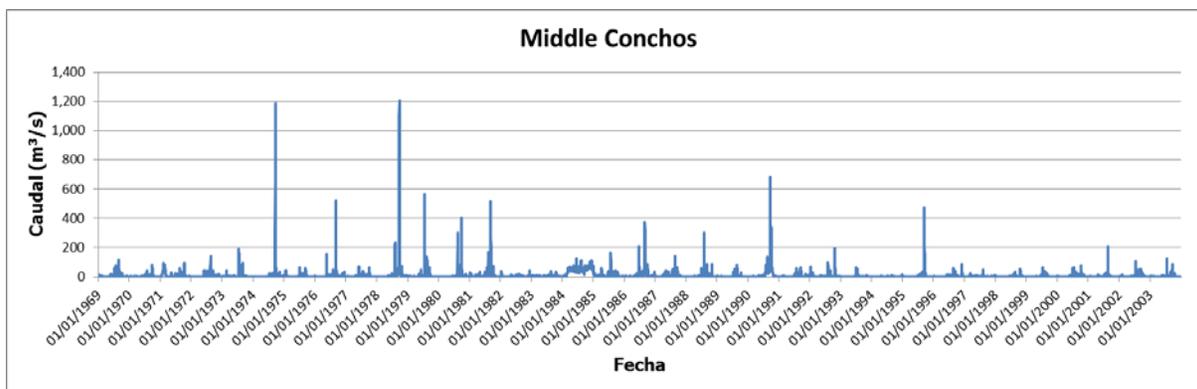


Figura 1.48 Subcuencas de simulación histórica

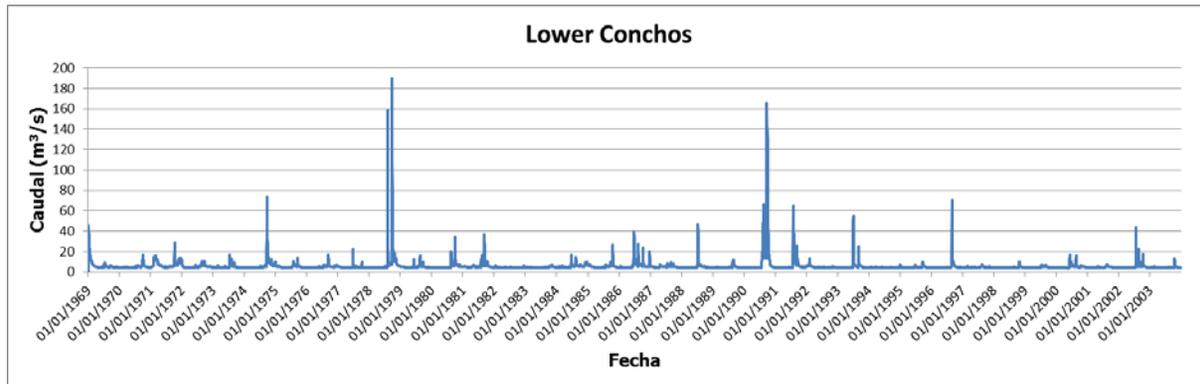


Figura 1.49 Subcuencas de simulación histórica

1.5.9 Pronóstico de escurrimientos

Con información de pronóstico de precipitación, temperaturas máximas y temperaturas mínimas para el periodo 01 de noviembre de 2012 al 04 de abril de 2013 se simularon las variables meteorológicas para obtener escurrimientos por cuenca propia en cada una de las subcuencas. Los resultados se muestran en la figura 1.50.

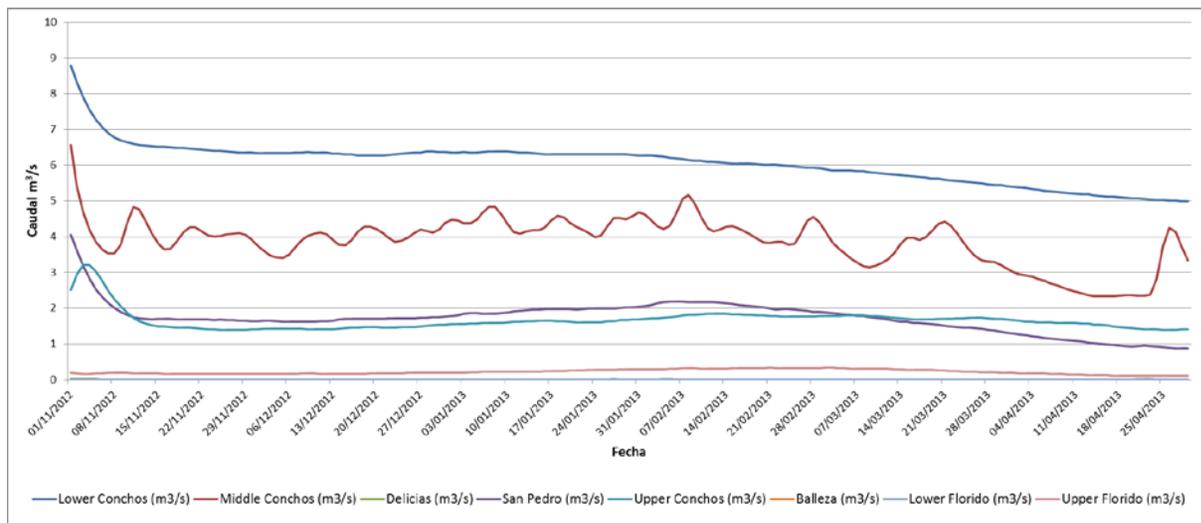


Figura 1.50 Simulación de pronóstico de meteorología

2. Modelación de la cuenca del río Yautepec

2.1 Conceptualización

El proyecto de modelación de la cuenca del río Yautepec, está integrado por dos modelos: Hidrológico e Hidráulico. Los dos modelos son desarrollados en forma independiente en distintas plataformas por lo que es necesaria la construcción de un vínculo que permita operarlos de manera coordinada, de tal forma que los resultados del modelo hidrológico son parte de los insumos del modelo de hidráulico. Los resultados del modelo hidrológico son colocados en hojas de cálculo, de donde son leídos por el modelo hidráulico.

El modelo hidrológico está desarrollado en las plataformas *Physitel* e *Hydrotel* (Ref. 3), las cuales calculan en conjunto el escurrimiento mediante un modelo hidrológico distribuido. El modelo es implementado en las tres subcuenas que integran el área de interés. Tiene asociado, un programa macro que transforma el formato *ASCII* a formato Excel que es legible por *HEC-RAS*.

El modelo hidráulico se desarrolló en la plataforma *HEC-RAS*. (Ref. 4). Es un modelo que procesa la información hidrométrica durante un evento extremo de precipitación, está vinculado con archivos de salida que contienen los datos iniciales y los resultados de la simulación.

2.1.1 Modelo hidrológico distribuido

De manera similar que la modelación hidrológica del río Conchos, el modelo hidrológico distribuido fue desarrollado en la plataforma *Hydrotel*. Simula a intervalo diario el escurrimiento directo, el escurrimiento retardado y el escurrimiento base en cientos de UHRH. La fisiografía de la cuenca se procesa previamente mediante la aplicación *Physitel*.

El modelo hidrológico considera para cada UHRH, mediante seis procesos de simulación, los datos fisiográficos, datos meteorológicos diarios y parámetros de procesos y de calibración. Cuatro de los procesos simulan el flujo vertical del agua en el suelo y los otros dos el flujo horizontal en superficie y cauces (Fig. 1.10). Para cada proceso existen algoritmos alternativos que se seleccionan en función de los datos disponibles (tabla 1.1), el resultado son series diarias en cualquier segmento de la red de corrientes de las variables principales del modelo. Asimismo, se generan diversos mapas temáticos, gráficos y tablas disponibles a exportar.

2.1.2 Modelo hidráulico

La conceptualización del modelo hidráulico consistió en abordar aspectos de discretización espacial en secciones transversales a cada 500 m dependiendo de las condiciones de la infraestructura existente y de caminos de acceso, en un tramo de seis kilómetros del río

Yautepec ubicado en la localidad del mismo nombre. Asimismo se consideraron las marcas de agua de los eventos históricos de inundación en ese tramo de seis kilómetros.

El levantamiento topográfico de cada una de las secciones transversales se realizaron tomando mediciones a cada dos metros de dependiendo de las condiciones topográficas y del caudal del río.

Los datos de las secciones transversales fueron modelados en el HEC-RAS, considerando paredes verticales en ambas márgenes del río una vez terminada la sección transversal.

2.1.3 Discretización temporal

El modelo hidráulico está desarrollado para simular un periodo de 12 horas, periodo en el cual se transita el hidrograma producto de la modelación hidrológica.

Los resultados se obtienen en reportes horarios, donde se tienen los niveles alcanzados por la inundación y que son comparados con los niveles registrados en las inundaciones históricas.

2.1.4 Discretización espacial

La discretización espacial se realizó en un tramo de río, iniciando en el punto de salida de la cuenca delimitada en el modelo hidrológico.

Desde el punto de vista hidrológico la discretización espacial se basó en las condiciones fisiográficas y se agruparon en relación a una estación hidrométrica:

- B-3 Nepantla (18302)
- Yautepec (18193)

La discretización espacial desde el punto de vista hidráulico se basó en 18 secciones transversales.

2.1.5 Desarrollo de la interfaz de Usuario

Para comunicar los modelos hidrológico e hidráulico se desarrolló una interfaz de usuario compuesta por tres secciones conceptuales: *Hydrotel*, *Hec-RAS* y Resultados.

Para lo cual, se desarrolló una aplicación de escritorio en el lenguaje de programación C#.net Framework 3.5, el cual permite establecer los vínculos de los modelos hidrológicos (*Hydrotel-Physitel*) e hidráulico (*HEC-RAS*), en esta interfaz de usuario se elaboró un mecanismo

homogéneo de navegación en las dos plataformas de aplicación, de tal manera que desde la pantalla de la interfaz se accede a la dos plataformas (*Hydrotel* y *HEC-RAS*).

Para esto fue necesario recopilar, analizar e interpretar los datos de entrada y de salida de los modelos *Hydrotel* y *HEC-RAS* (Fig. 2.1), esto con el fin de determinar la manera de enlazar estos dos modelos, de tal manera que el usuario pueda de forma rápida determinar zonas de riesgo en cuencas previamente modeladas y llevar a cabo medidas de prevención para mitigar los daños que estos fenómenos naturales pudieran ocasionar.

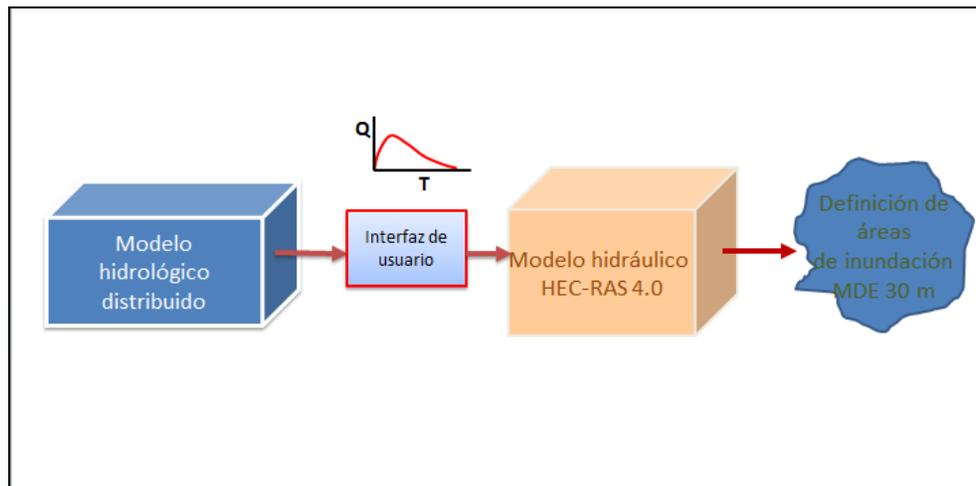


Figura 2.1 Modelo coordinados *Hydrotel/HEC-RAS*

2.2 Recopilación y análisis de información

La recopilación de información de manera similar en la modelación de la cuenca del río Conchos, esta consistió en formar una base de datos sobre rasgos geográficos y series de tiempo.

2.2.1 Información geográfica

La base de datos geográfica contiene rasgos vectoriales y matriciales en la proyección Cónica Conforme de Lambert, *Datum* WGS 84.

2.2.2 Modelo digital de elevaciones

Se obtuvo el modelo digital de elevaciones (MDE) para la cuenca completa del río Yautepec (Ref. 1), con resolución de celda de un segundo, cerca de 30 metros (Fig. 2.2).

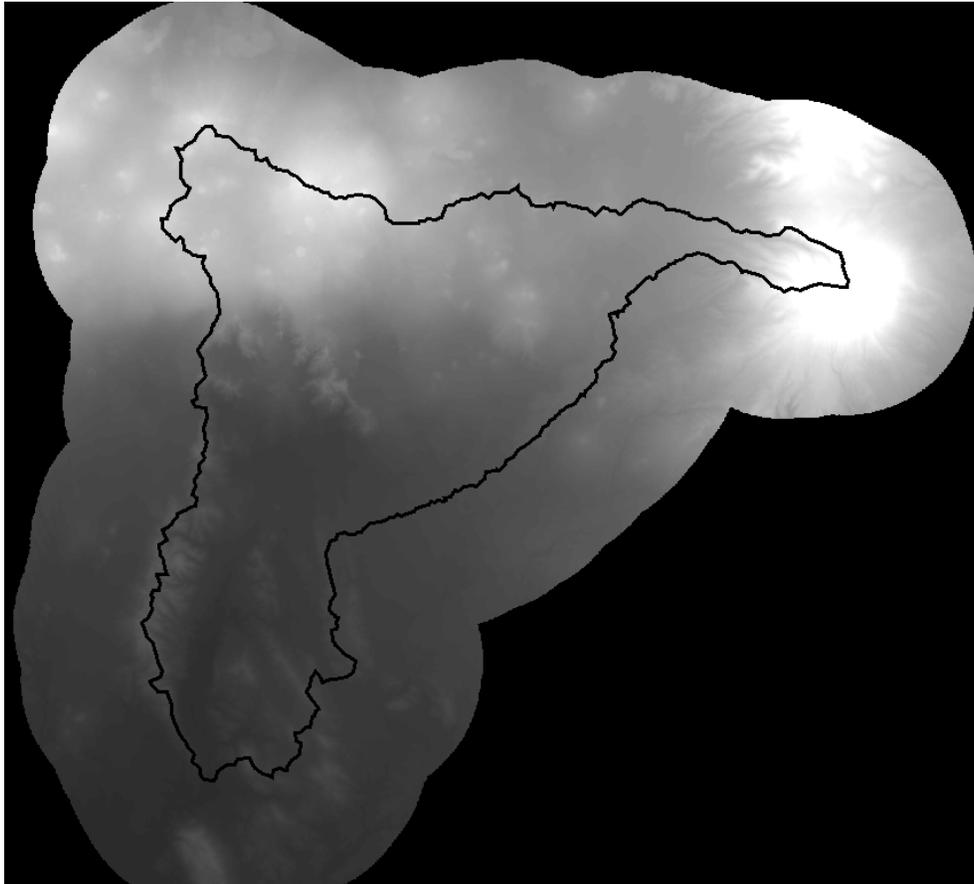


Figura 2.2 Modelo Digital de Elevaciones del SRTM de la cuenca del río Yautepec

El MDE se utilizó en la generación de la fisiografía de la cuenca en *Physitel*. Comprende en su totalidad el área de la cuenca junto con un margen de 10 km a partir del parteaguas generado por el *Phystel* para la cuenca.

2.2.3 Parteaguas y subcuencas

La cuenca del río Yautepec cuenta con tres subcuencas las cuales fueron definidas con base a los puntos de control establecido. En la figura 2.3 se muestra las subcuencas con el nuevo trazo del parteaguas obtenido a partir del MDE en *Physitel*.

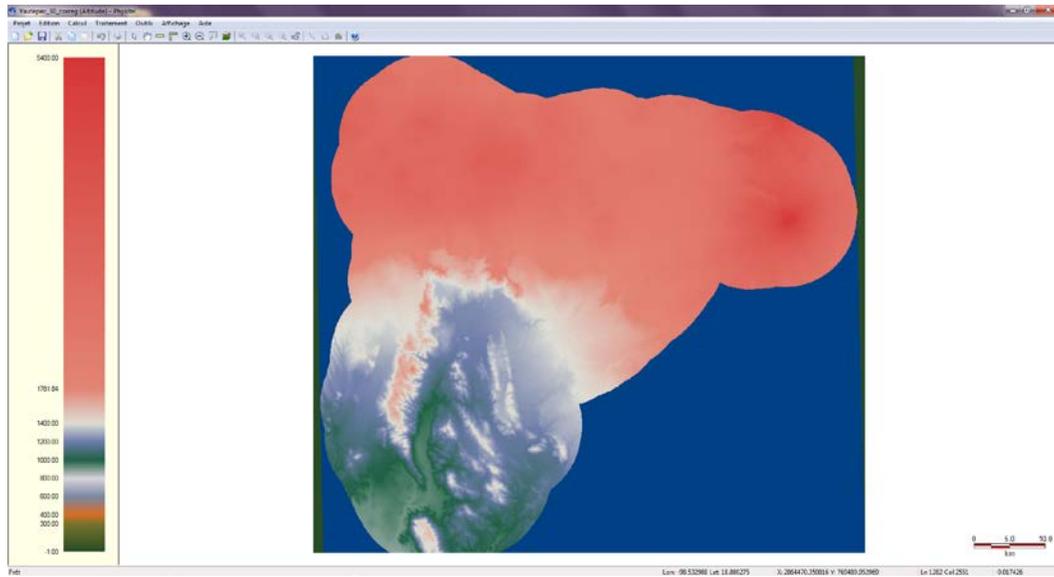


Figura 2.3 Nuevo parteaguas generado en Physitel y subcuencas hidrológicas de la cuenca del río Yautepéc

2.2.4 Red de drenaje

La red de drenaje para la cuenca del río Yautepéc (Fig. 2.4) es producto de cartas vectoriales escala 1:50,000 del *INEGI*.

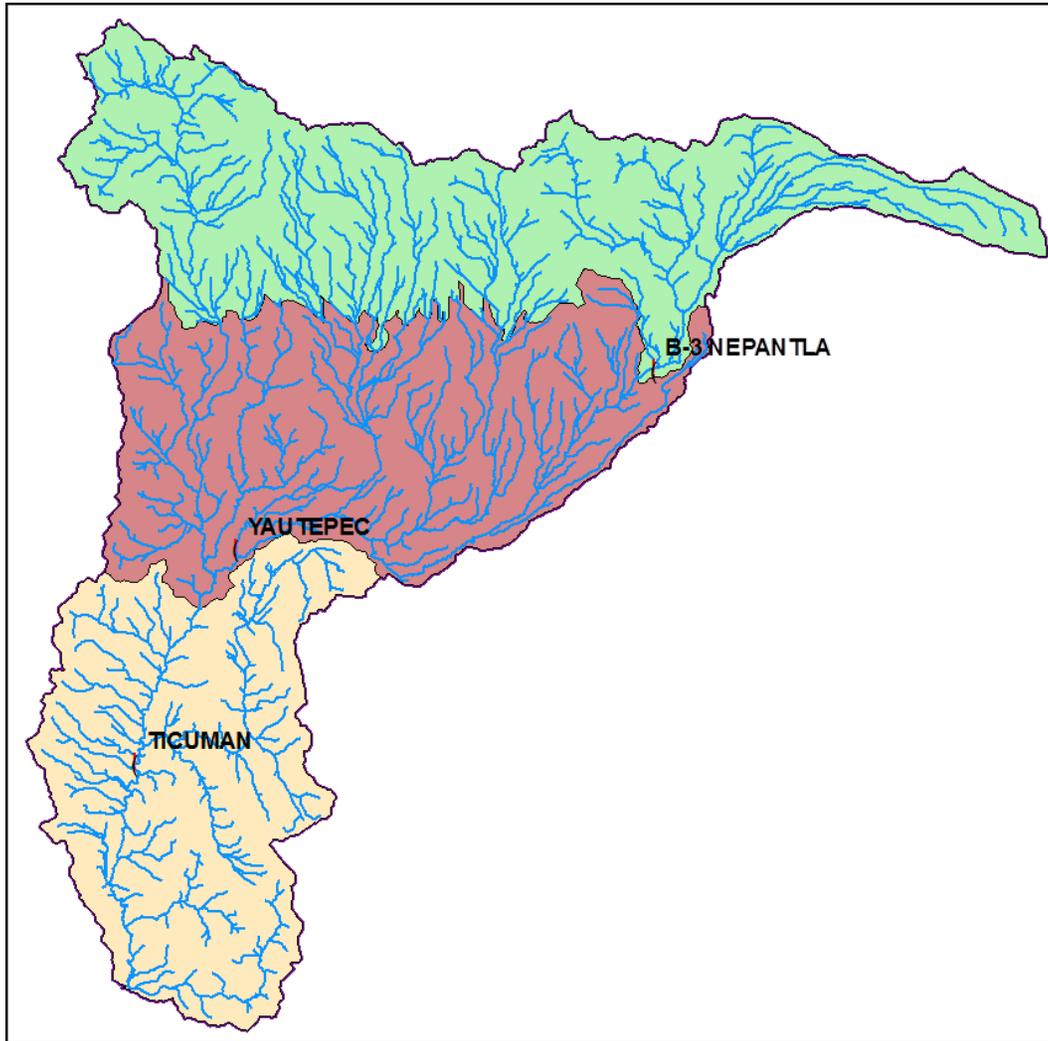


Figura 2.4 Red hidrográfica vectorial escala 1:50,000 de la cuenca del río Yautepec

La hidrografía fue utilizada para indicarle a la aplicación de *Physitel* la correcta trayectoria que sigue cada una de las corrientes (Ref. 2). Se realizó una edición a la red mostrada en la figura para utilizarla en *Physitel*. La edición consistió en quitar y conectar corrientes tributarias hasta obtener una red de drenaje compuesta solamente por corrientes principales, para ello se toma como base la red de drenaje que se genera en *Physitel* previamente. La red resultante es utilizada en el *Physitel* (Fig. 2.5).

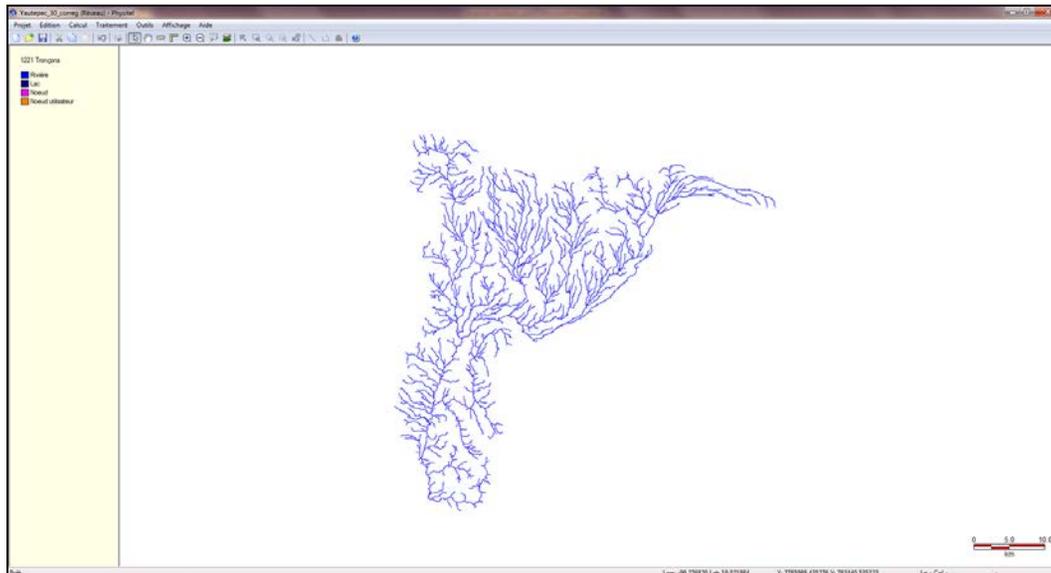


Figura 2.5 Red de drenaje escala 1:50,000 editada tomando como base la red que se genera a partir de MDE en *Physitel*

2.2.5 Uso de suelo

La cobertura de uso de suelo (Fig. 2.6) se obtuvo de la cartografía escala 1:250,000, generada por el Instituto de Geografía de la *UNAM* para el Inventario Nacional Forestal 2000, y modificada y homologada en sus clases a dos sistemas de clasificación más por el Instituto Nacional de Ecología (*INE*).

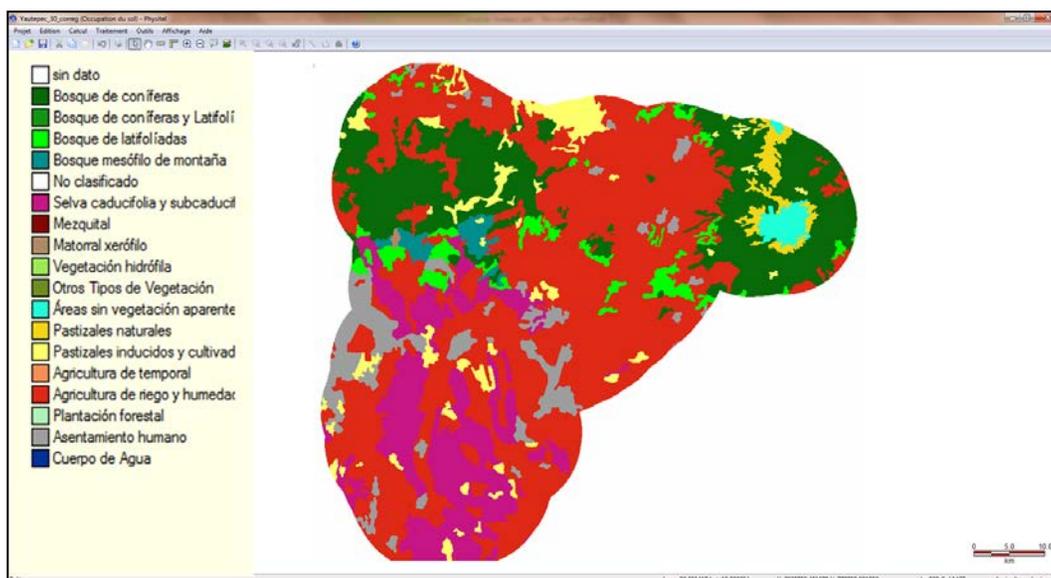


Figura 2.6 Uso de suelo de la cuenca de río Yautepac según el sistema de clasificación de cobertura vegetal ajustados por el *INE*

La capa vectorial de uso de suelo (Fig. 2.6), se incorpora al modelo a través de *Physitel* en formato *.txt* obtenida por un proceso de transformación a formato matricial. El resultado es una malla en la que cada celda contiene una sola clase de uso de suelo (Fig. 3.12).

2.2.6 Edafología

La edafología de la cuenca del río Yautepec se obtuvo a partir del mosaico de las cartas edafológicas producido por el INEGI escala 1:250,000. Los tipos de suelo se clasificaron siguiendo los criterios establecidos por la *FAO/UNESCO* en 1968 en el *World Reference Base (WRB)* y modificado por DETENAL en 1970. En México, el sistema de clasificación consta de 25 grandes grupos de clases deducidas a partir de la génesis de los suelos. Los grupos se expanden a un segundo nivel en el que al nombre se les añade un adjetivo para hacer más precisa la clasificación. Las cartas también constan de información sobre la fase física, química y textural ésta última se reduce a establecerla como gruesa media y fina. En la figura 2.7 se muestra la textura del suelo de la cuenca.

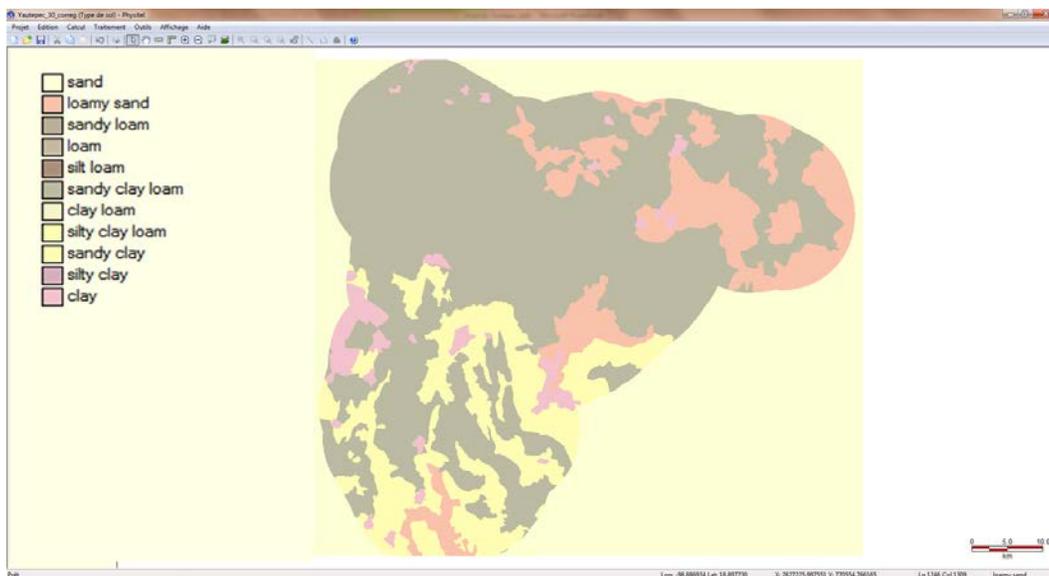


Figura 2.7 Capa textura de suelo para la cuenca del río Yautepec

Al igual que para el uso de suelo, la textura del suelo es importada a *Physitel* a través del formato *.txt*, producto de la conversión a formato matricial de las clases texturales de la edafología. Finalmente, cada celda del mapa matricial contiene una textura representativa del suelo considerando el rodal dominante.

2.3 Series de tiempo

Se prepararon las series históricas completas de temperatura mínima, temperatura máxima y precipitación diarias en formato legible por *Hydrotel*. Asimismo, se prepararon las series históricas completas de escurrimiento diario en tres estaciones de control ubicadas en o cerca de la salida de las subcuencas.

2.3.1 Estaciones hidrométricas

Se identificaron 16 estaciones hidrométricas en la cuenca del Yautepec, todas se ubicaron geográficamente (Fig. 2.8). Se seleccionaron tres estaciones de calibración para el modelo hidrológico.

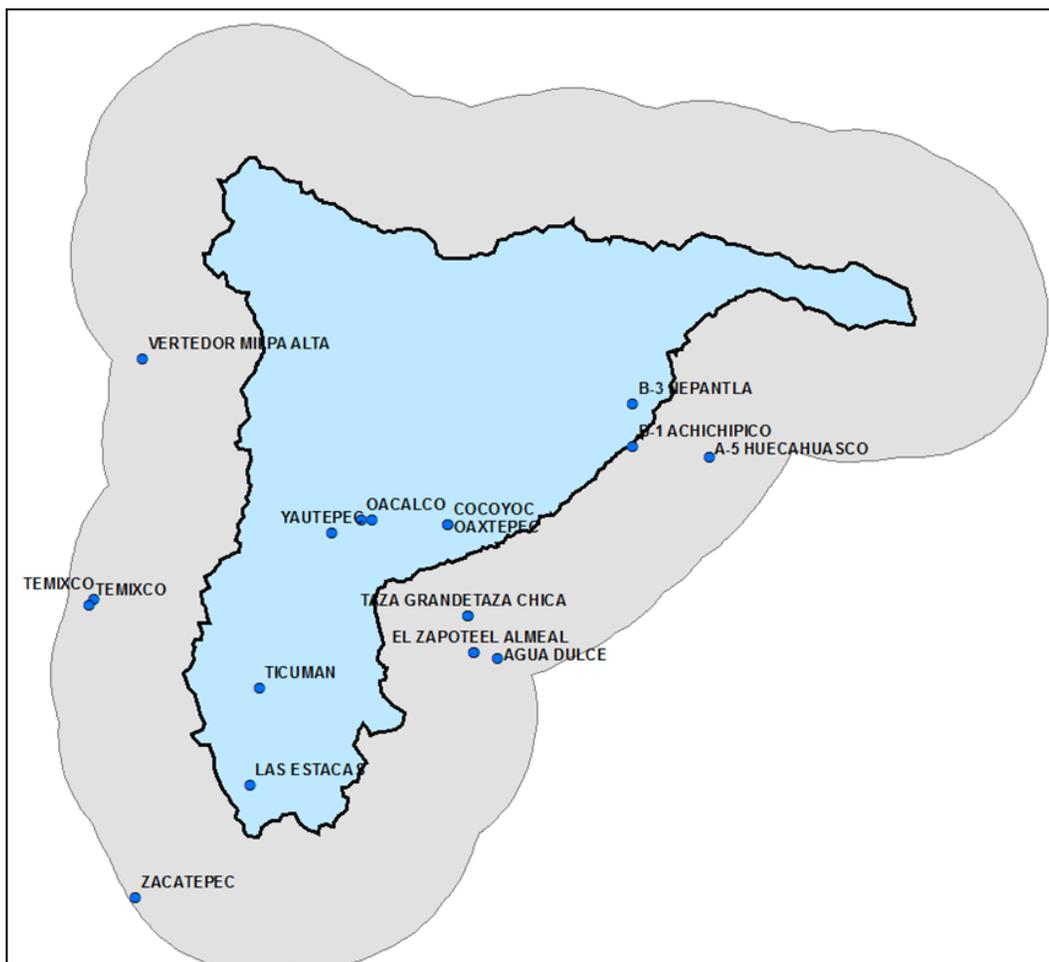


Figura 2.8 Ubicación de las estaciones hidrométricas en la cuenca del río Yautepec

2.3.2 Estaciones meteorológicas

De un total de 46 estaciones meteorológicas (ERIC III) que comprenden el área de estudio, con margen de 10 km a partir del parteaguas (Fig. 2.9), éstas se utilizaron para calibrar el modelo hidrológico en el período de 1980-1984.

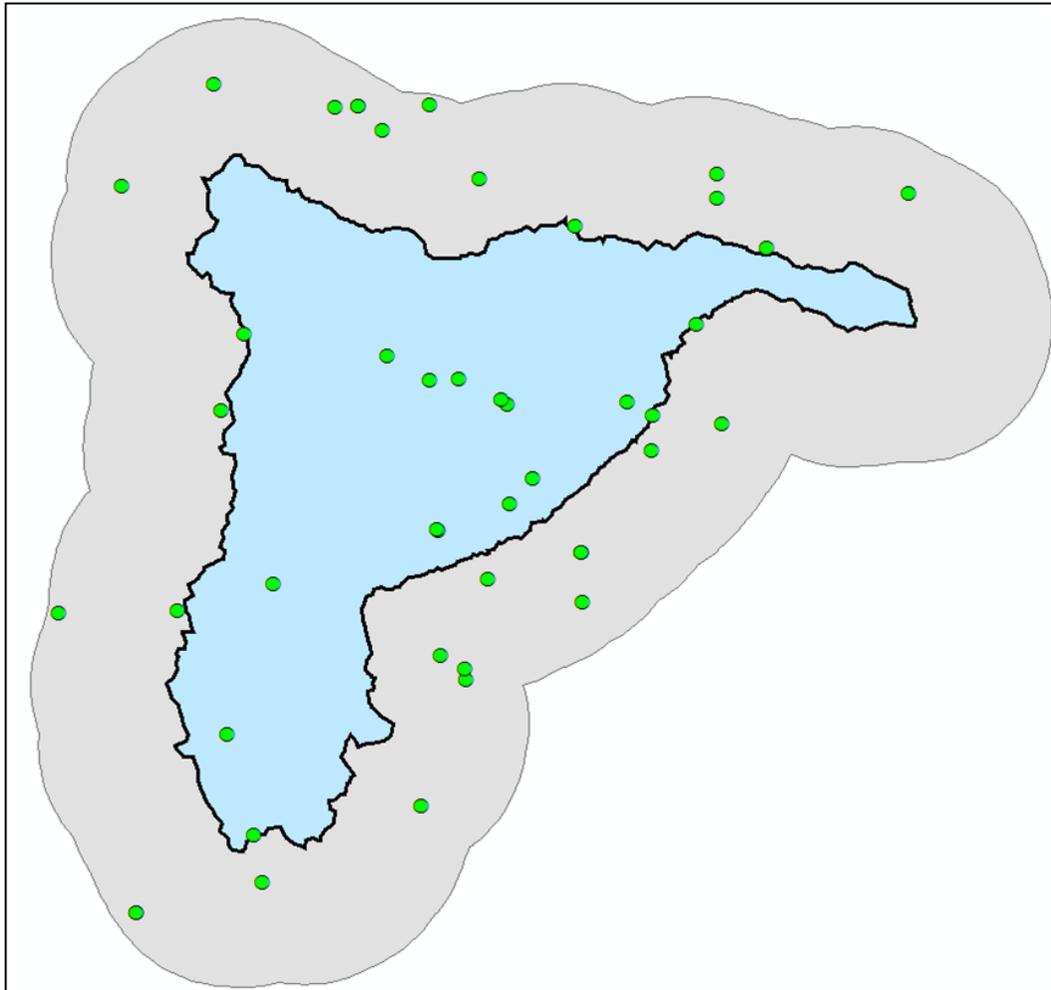


Figura 2.9 Cobertura de estaciones meteorológicas

Se hizo un análisis de las estaciones climatológicas, y se identificó que la mayor densidad de registros se encontró en el periodo de 1980-1984 y la tendencia en cuanto a número de estaciones va sensiblemente a la baja y la cantidad de registros en cada estación su variación se mantiene constante (fig. 2.10 a y b).

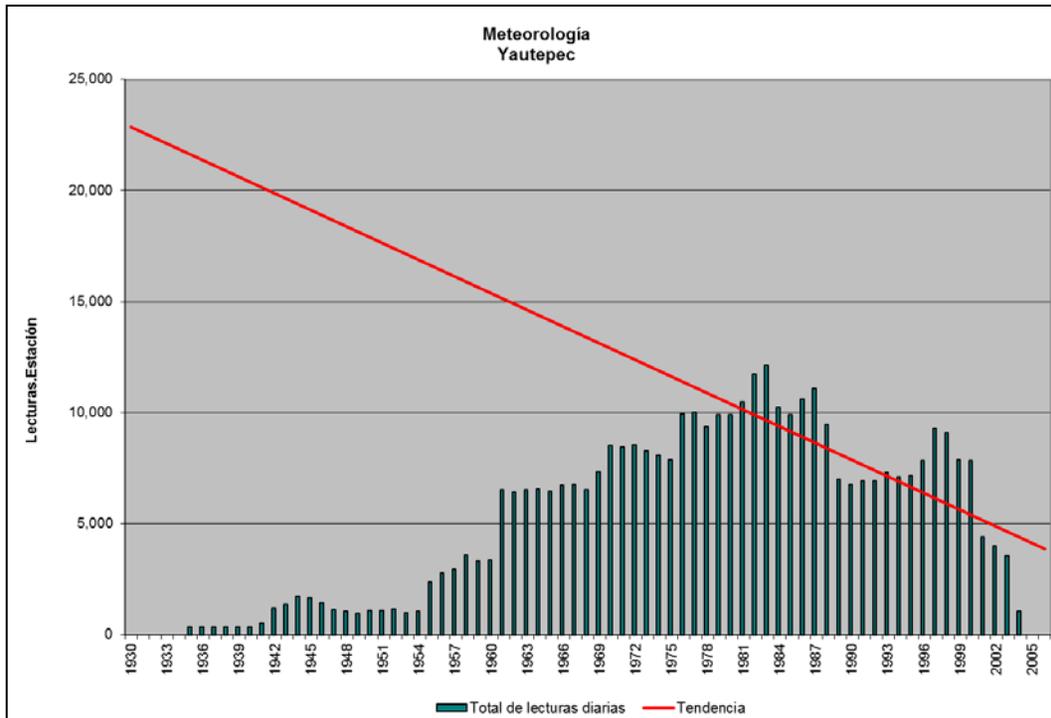


Figura 2.10a Análisis de cantidad de estaciones meteorológicas

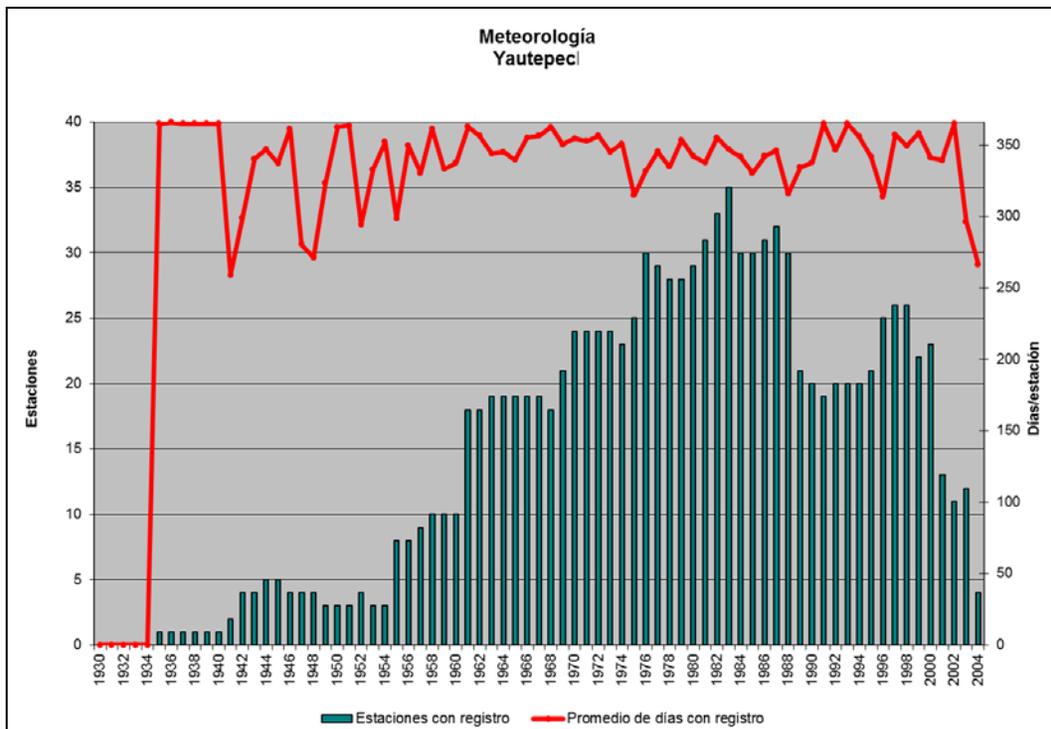


Figura 2.10b Análisis de registros en estaciones meteorológicas

2.4 Proyecto Physitel

Como ya se ha comentado, el modelo hidrológico se desarrolló en dos plataformas, *Physitel* e *Hydrotel*, donde primero se debe desarrollar el proyecto *Physitel*, que consta de 13 etapas a través de las cuales se definen las características de la cuenca, las cuales son exportadas al *Hydrotel* para la modelación hidrológica.

-  Etapa 1. Importación del modelo digital de elevación
-  Etapa 2. Determinación y edición de pendientes
-  Etapa 3. Importación y edición de la red vectorial de corrientes impuesta
-  Etapa 4. Determinación y edición de la red matricial de corrientes impuesta
-  Etapa 5. Determinación y edición del modelo digital de elevación modificado
-  Etapa 6. Determinación y edición de las direcciones de flujo
-  Etapa 7. Determinación del parteaguas de la cuenca
-  Etapa 8. Determinación de la red de corrientes modelada
-  Etapa 9. Reorientación de las celdas alrededor de los nodos
-  Etapa 10. Determinación y edición de las unidades hidrológicas
-  Etapa 11. Importación y edición de los usos de suelo
-  Etapa 12. Importación y edición de los tipos de suelo
-  Etapa 13. Exportación del proyecto *Physitel* al formato *Hydrotel*

2.4.1 Importación del modelo digital de elevación

Esta etapa consistió en incorporar a *Physitel* el modelo digital de elevaciones (Fig. 2.11), a partir del cual se definió la fisiografía de la cuenca. Para fines de *Physitel* y dado que este no cuenca con la proyección *Lambert Conformal Conic (LCC)* con parámetros Mexicanos, se utilizó la *LCC* pero con parámetros de Norteamérica.

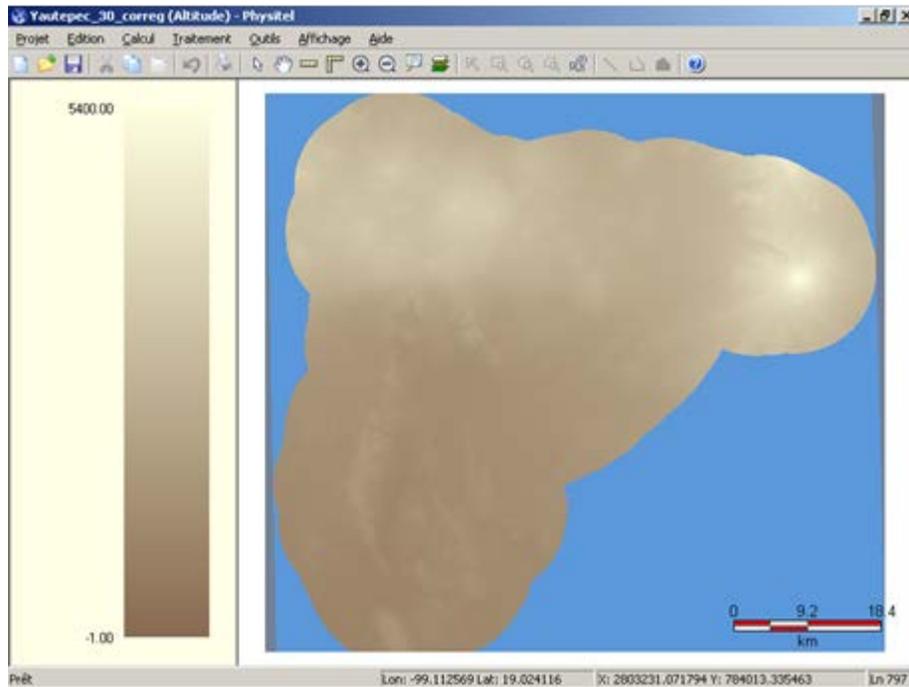


Figura 2.11 Modelo digital de elevaciones de la cuenca del río Yautepec, etapa 1 de Physitel

2.4.2 Determinación y edición de pendientes

Las pendientes son determinadas por *Physitel* en función de la elevación del terreno (MDE) y la distancia, es decir, el *Physitel* determina la pendiente para cada celda (Fig. 2.12) tomando en cuenta las elevaciones de las celdas vecinas.

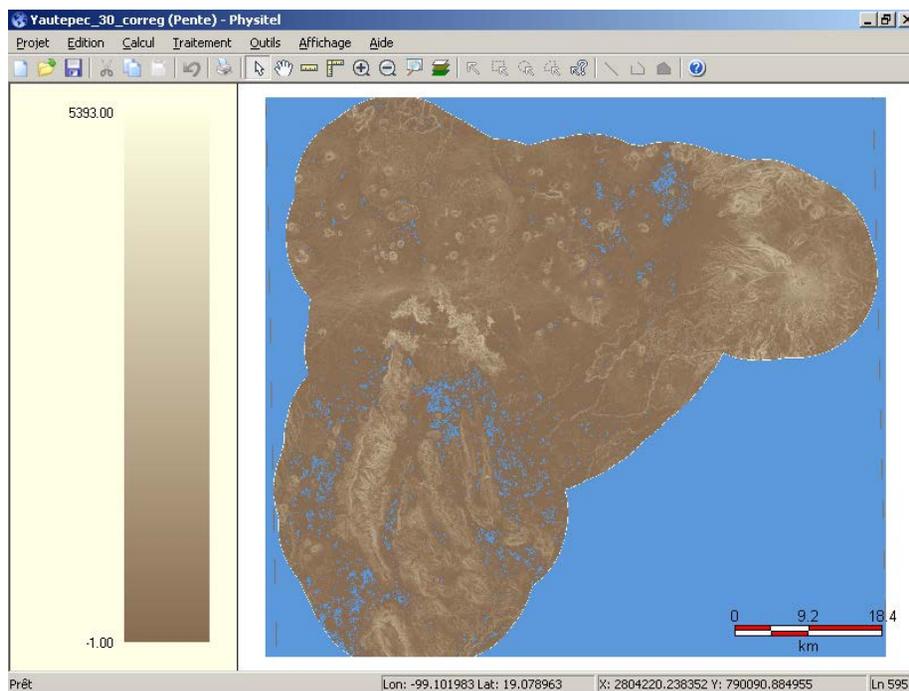


Figura 2.12 Pendientes generadas en la etapa 2 de Physitel a partir del MDE

2.4.3 Importación y edición de la red vectorial de corrientes impuesta

Esta etapa consiste en incorporar la red hidrográfica bajo un formato vectorial (Fig. 2.13). Procesado en el apartado 1.1.6 (Fig. 1.10). Y a partir de la cual *Physitel* determinará una red matricial siguiendo la correcta ubicación de la red hidrológica vectorial impuesta.

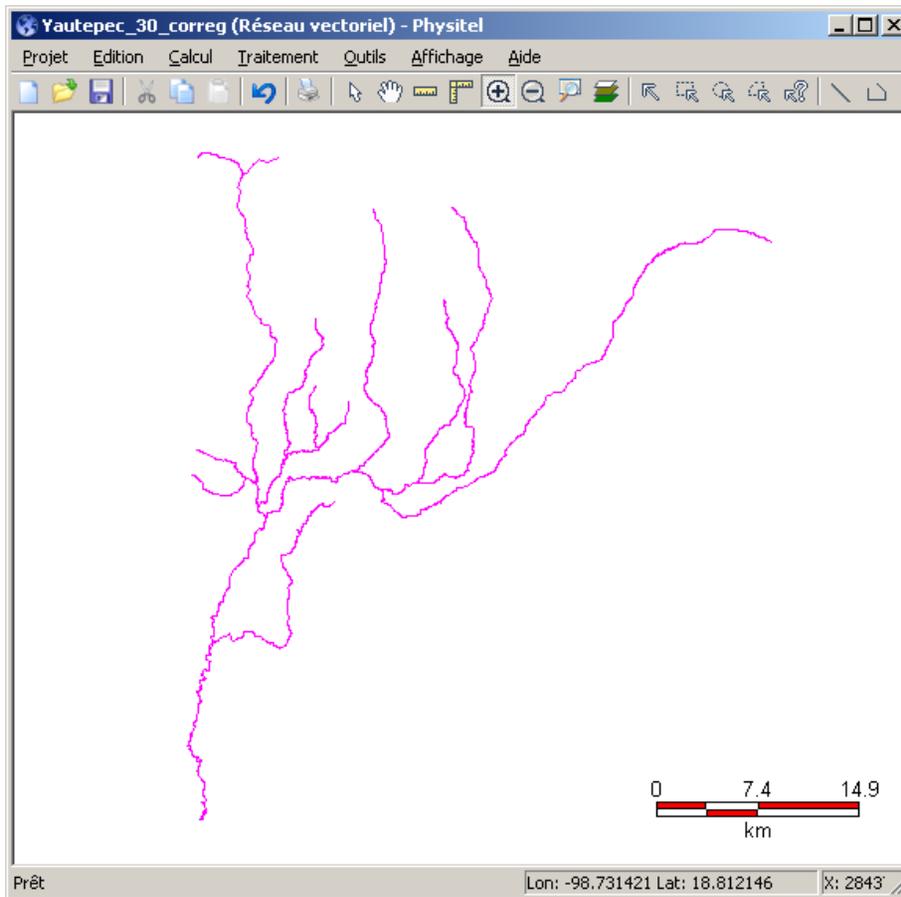


Figura 2.13 Red vectorial escala 1:50,000 dentro de *Physitel*

2.4.4 Determinación y edición de la red matricial de corrientes impuesta

La determinación y edición de la red matricial de corrientes (Fig. 2.14) consistió en obtener una representación matricial (Malla) de la red hidrográfica vectorial impuesta, es decir la red matricial resultante es en realidad el quemado de la red vectorial en el MDE a nivel celda.

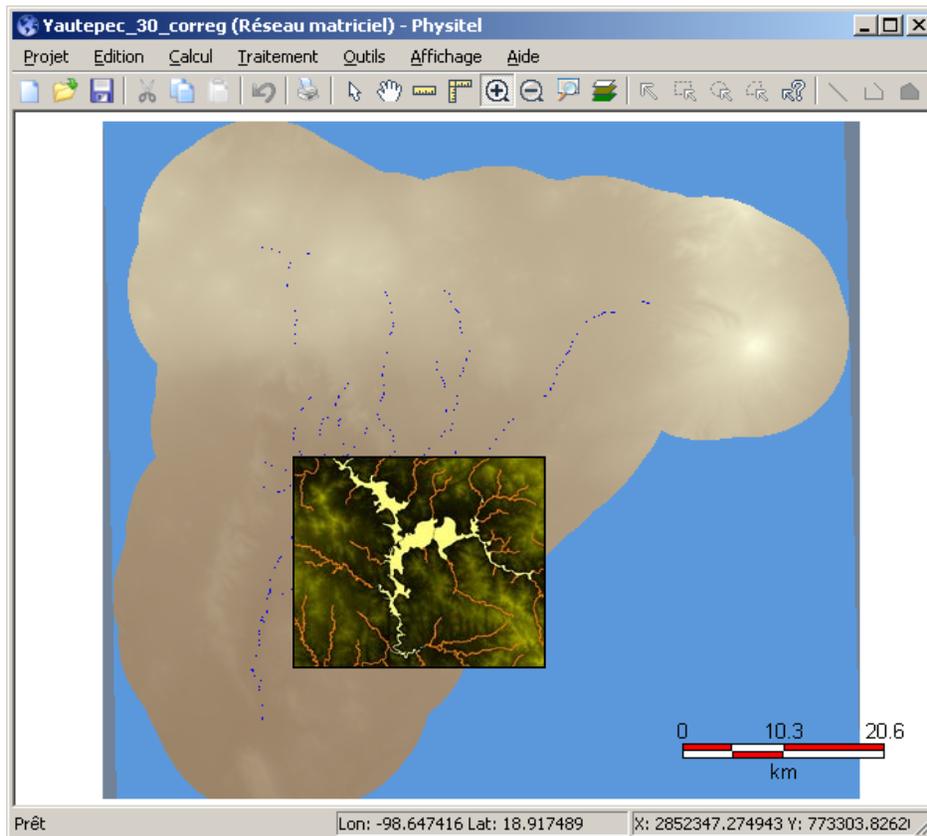


Figura 2.14 MDE y la red matricial generada a partir de la red vectorial 1:50,000 (INEGI)

2.4.5 Determinación y edición del modelo digital de elevación modificado

Esta etapa consistió en modificar el MDE por medio de un hundimiento de las elevaciones registradas sobre y en la periferia de la red matricial generada (Fig. 2.15), con el fin de dar una afluencia de los escurrimientos hacia las corrientes.

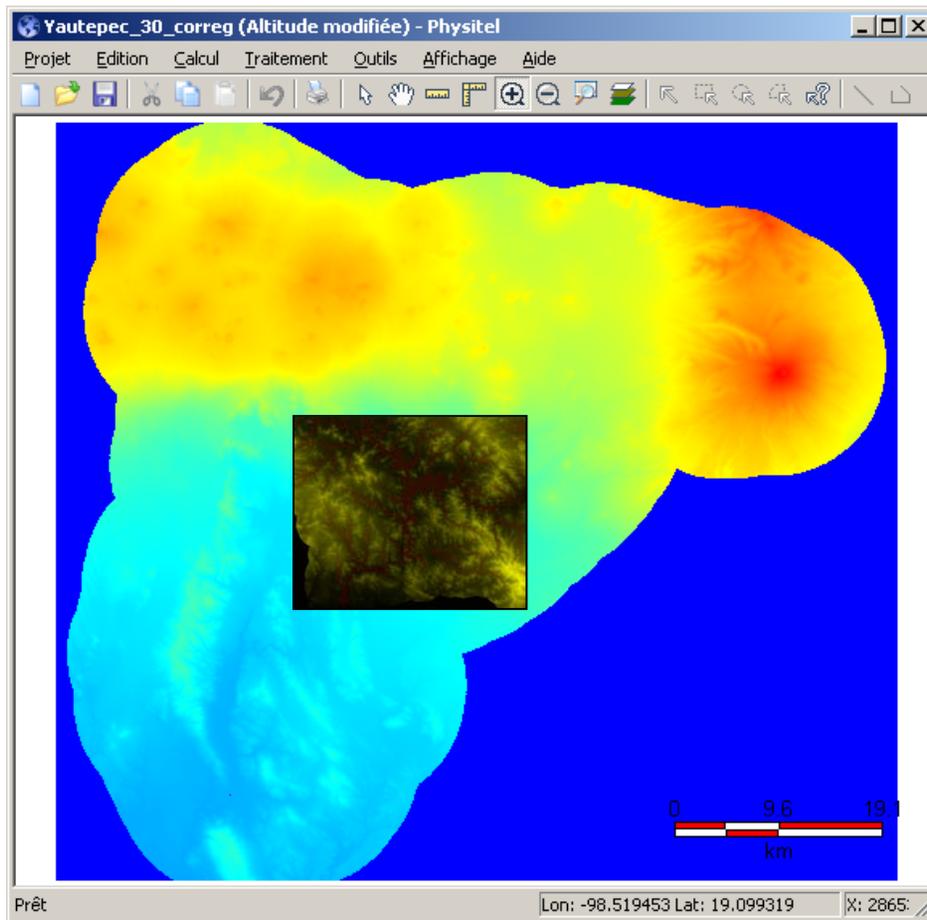


Figura 2.15 Muestra el área de influencia modificada en el MDE en función de las corrientes y cuerpos de agua

2.4.6 Determinación y edición de las direcciones de flujo

En esta etapa *Physitel* le asigna una dirección de flujo a cada celda del MDE que constituye el área de estudio (Fig. 2.16). La dirección resultante para cada celda es producto del peso de la pendiente y así como la red matricial.

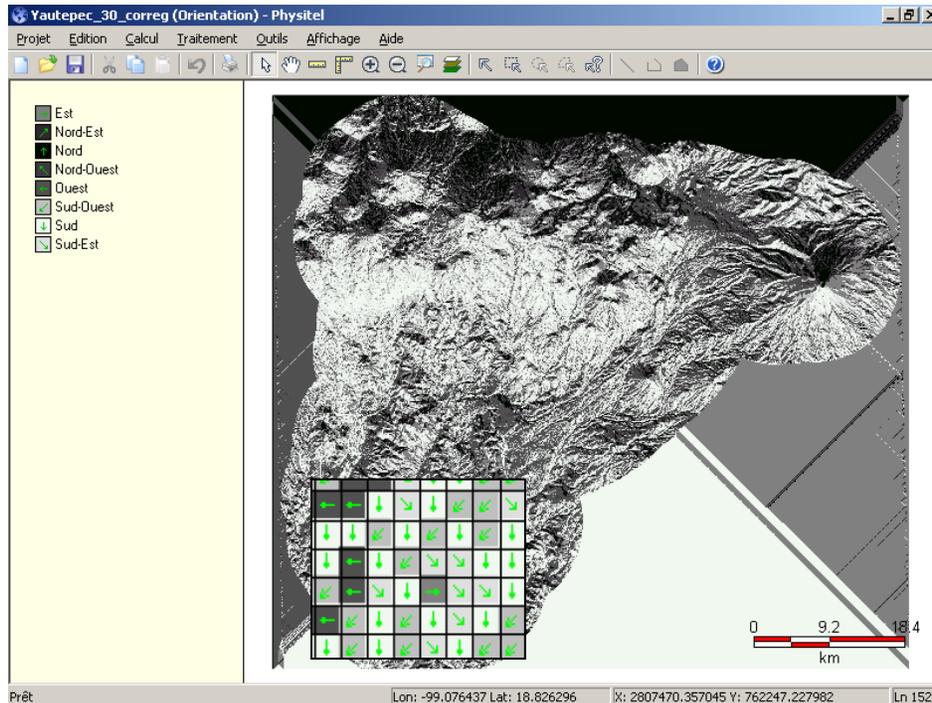


Figura 2.16 Muestra el resultado del cálculo realizado en la etapa 6, así como un acercamiento donde se puede observar las direcciones obtenidas a nivel celda

La edición de las direcciones consistió básicamente en asignar un sentido a las celdas indeterminadas que son celdas a las cuales *Physitel* no les dio una dirección de flujo.

2.4.7 Determinación del parteaguas de la cuenca

Con base en la conectividad de direcciones de flujo entre celdas *Physitel* determina el área de drenaje de la cuenca (parteaguas). Asimismo, se obtuvo la red de corriente total de la cuenca del Yautepec (Fig. 2.17), esta red contiene ya todo los cuerpos de aguas seleccionados con anterioridad (apartado 3.1.6) para la modelación en *Hydrotel*.

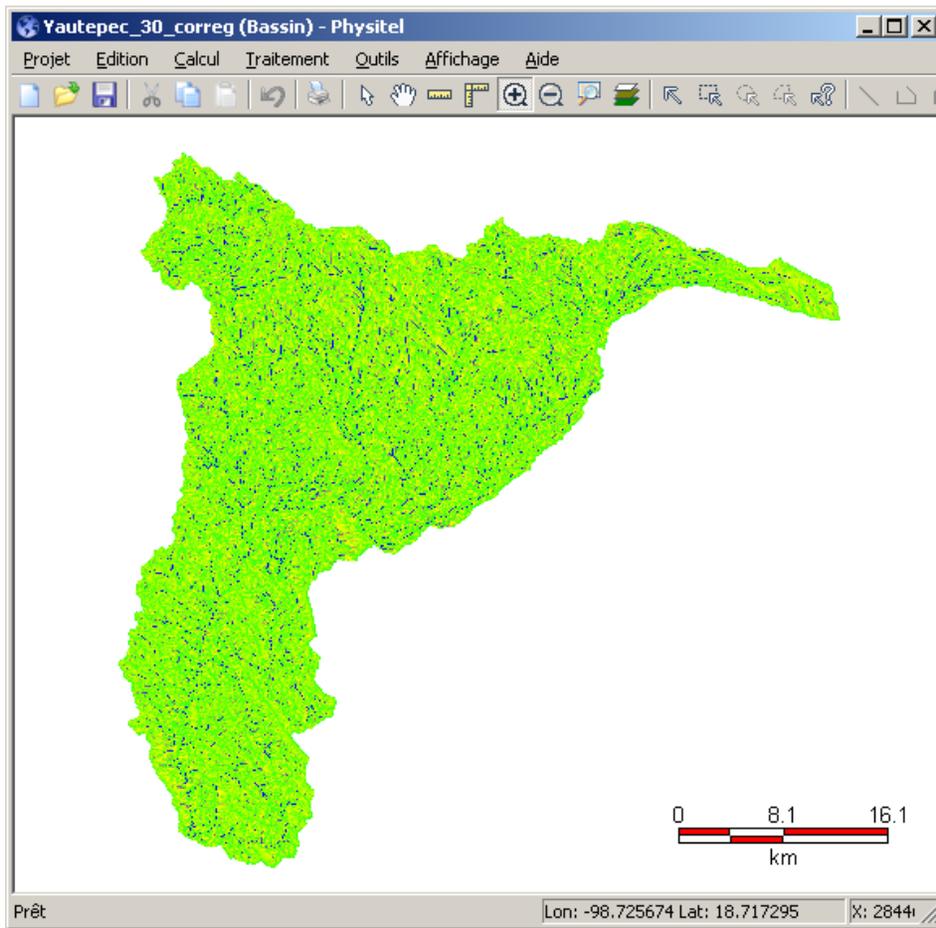


Figura 2.17 Parteaguas y red de corrientes resultado de la etapa 7 de Physitel

2.4.8 Determinación de la red de corrientes modelada

La etapa ocho de *Physitel* permitió obtener la red a modelar (Fig. 2.18), lo anterior se realizó usando un umbral de densidad de corrientes de 500.

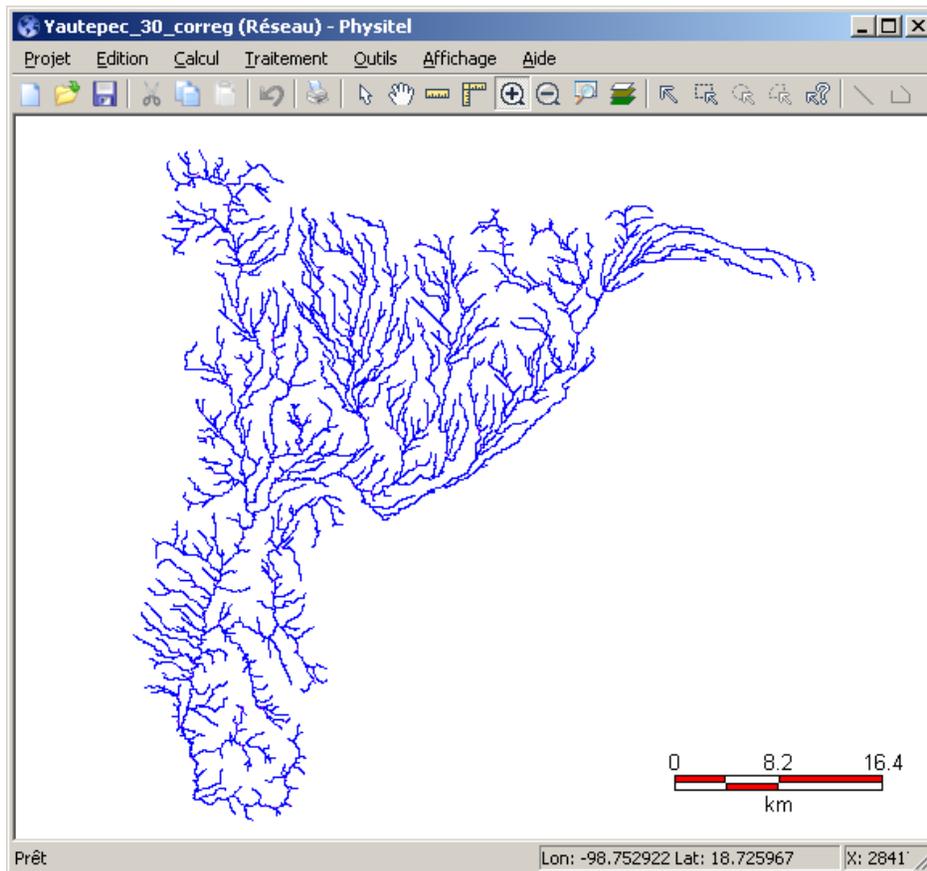


Figura 2.18 Muestra la red de corrientes a modelar

Para la calibración en cada una de las subcuencas (Fig. 1.3) se insertaron 12 puntos de control (Fig. 2.19) y se volvieron a determinar las corrientes con la incorporación de estos nuevos nodos.

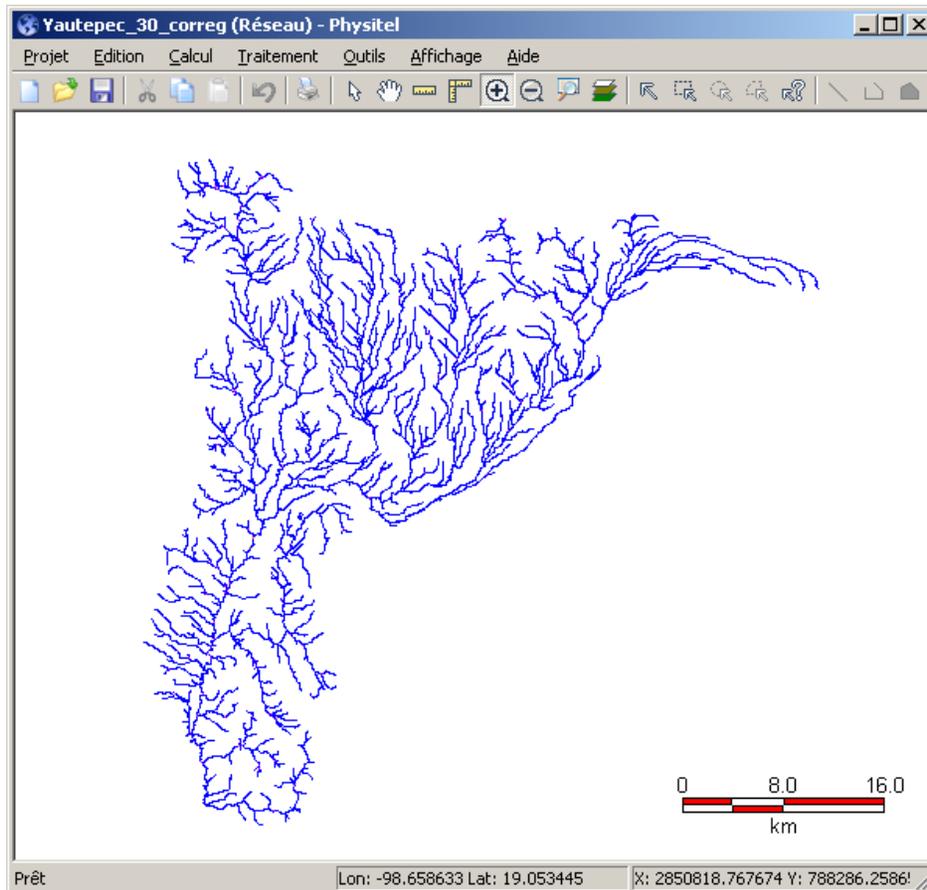


Figura 2.19 Dos puntos de control de las estaciones hidrométricas insertados en la red de corriente

2.4.9 Determinación y edición de las unidades hidrológicas

Las unidades hidrológicas relativamente homogéneas (UHRH) son áreas de drenaje correspondientes a una corriente y a un nodo, éstas las relaciona y clasifica el sistema *Physitel* estableciendo una numeración en sentido horario de la corriente, comenzando la enumeración sobre la margen izquierda de las corrientes. Para la cuenca del Yautepec se obtuvieron 1,885 unidades hidrológicas relativamente homogéneas (Fig. 2.20).

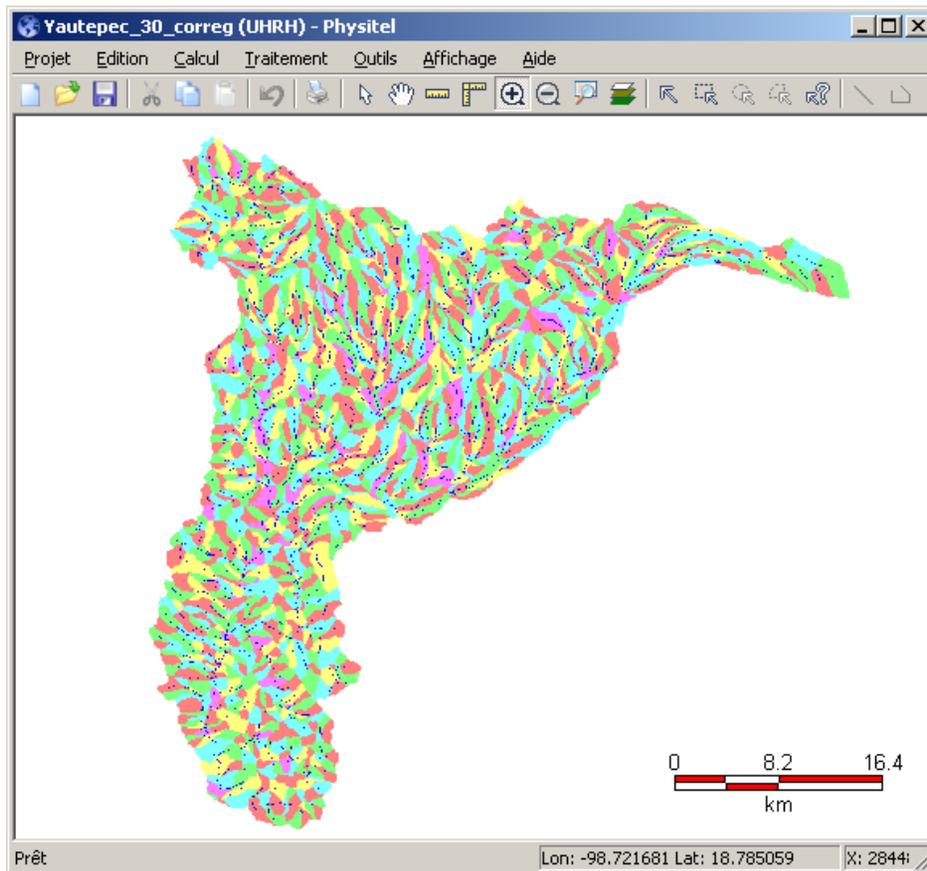


Figura 2.20 Muestra las UHRH obtenidas para la cuenca del Yautepec

2.4.10 Importación y edición de los usos de suelo

La incorporación de los usos de suelo a *Physitel* fue a través del archivo ".txt", el cual cuenta con 19 clases de usos de suelo de la cuenca Yautepec de acuerdo con la clasificación del INE. *Physitel* calcula la ocupación del suelo a nivel de *UHRH* (Fig. 2.21).

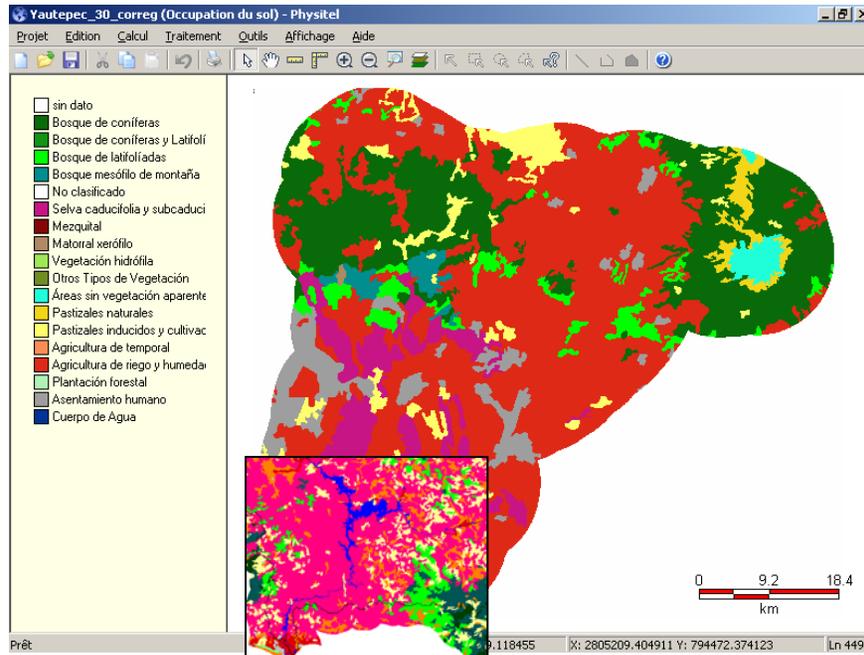


Figura 2.21 Muestra los diferentes usos de suelo en la cuenca del río Yautepec

2.4.11 Importación y edición de los tipos de suelo

La edafología de la cuenca del Yautepec en ambiente *Physitel* es producto del archivo con extensión ".txt", mencionado en el apartado 1.1.10. Este archivo cuenta con 11 clases edafológicas producto de la clasificación que maneja *Physitel* (Fig. 2.22). En esta etapa *Physitel* calcula en porcentajes el tipo de suelo para cada UHRH.

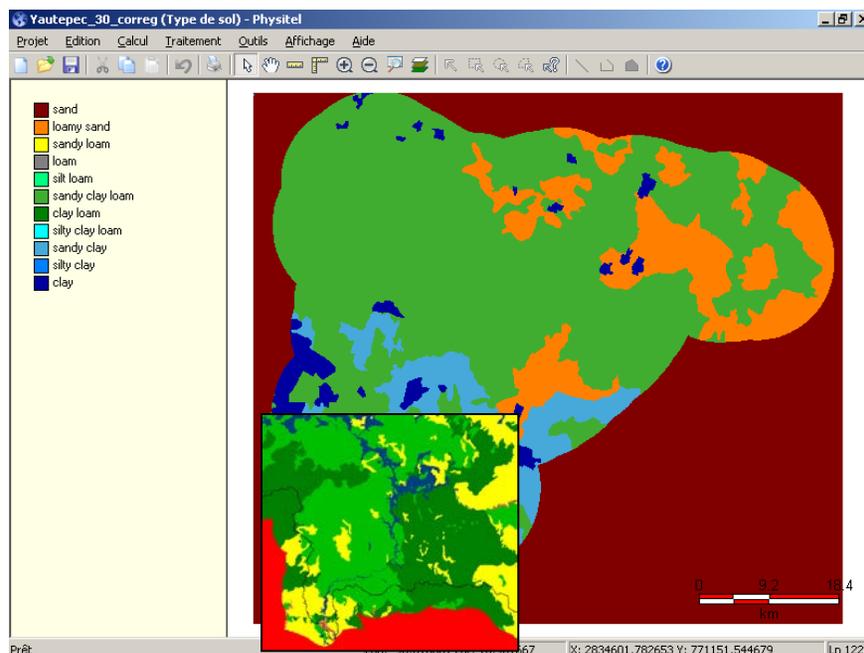


Figura 2.22 Muestra la edafología de la cuenca del río Yautepec en el ambiente *Physitel*

2.4.12 Exportación del proyecto *Physitel* al formato *Hydrotel*

La exportación es la última etapa del *Physitel*, consiste en cambiar la información (Fig. 2.23) al formato *Hydrotel*. Los archivos transformados son exportados a una carpeta llamada *Hydrotel*.

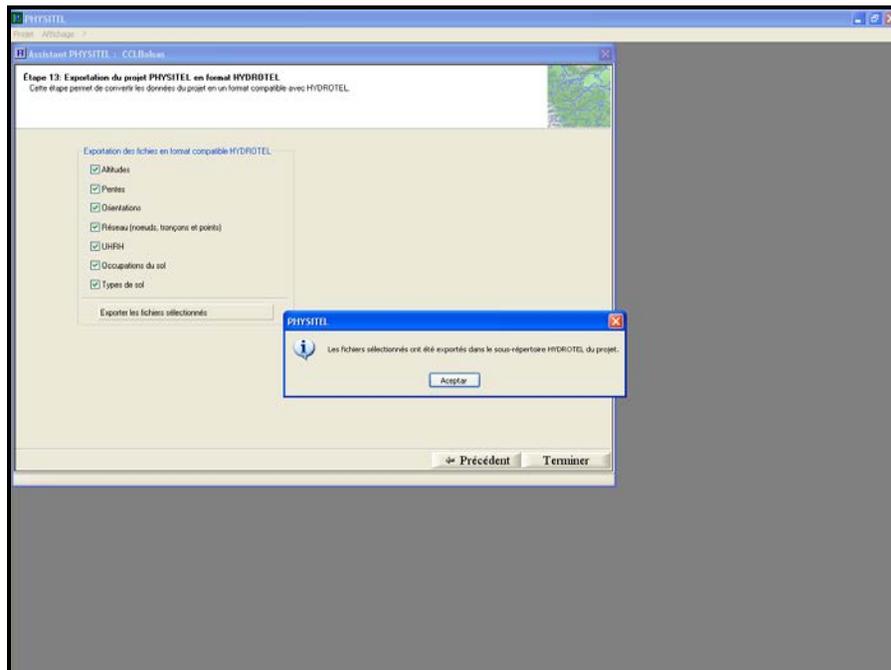


Figura 2.23 Muestra las coberturas exportadas a formatos *hydrotel*

2.5 Proyecto *Hydrotel*

En la creación del proyecto *Hydrotel* es necesario importar los archivos procesados por *Physitel*, es decir, la fisiografía de la cuenca. Así también, archivos que contienen información meteorológica, hidrométrica e información de vegetación. Todo esto con el fin de representar de una forma más aproximada las condiciones naturales de la cuenca y aprovechar al máximo la información disponible.

2.5.1 Importación de archivos *Physitel*

Los archivos que se exportaron de formato *Physitel* a formato *Hydrotel* son: UHRH, altitudes, orientaciones, pendientes, nodos, tramos, puntos, edafología, tipo de suelo y propiedades hidráulicas. En la figuras 2.24 y 2.25 se muestra la interfaz en donde se indica la ubicación de los archivos exportados por *Physitel*.

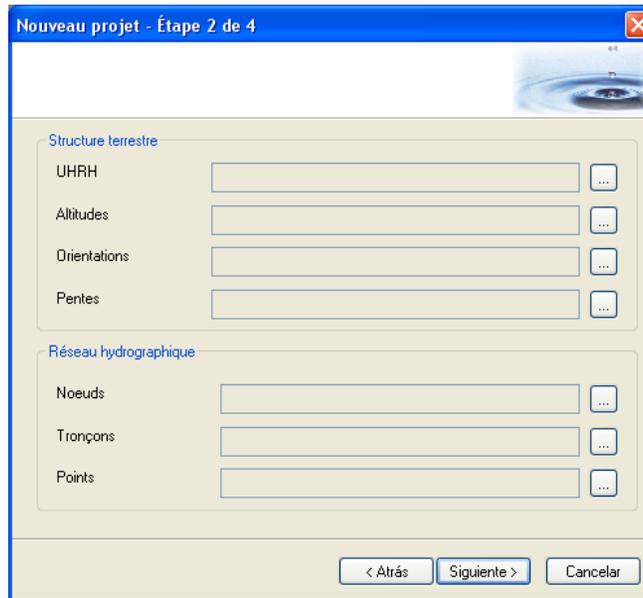


Figura 2.24 Interfaz de importación de archivos de fisiografía de la cuenca para el montaje del proyecto en Hydrotel

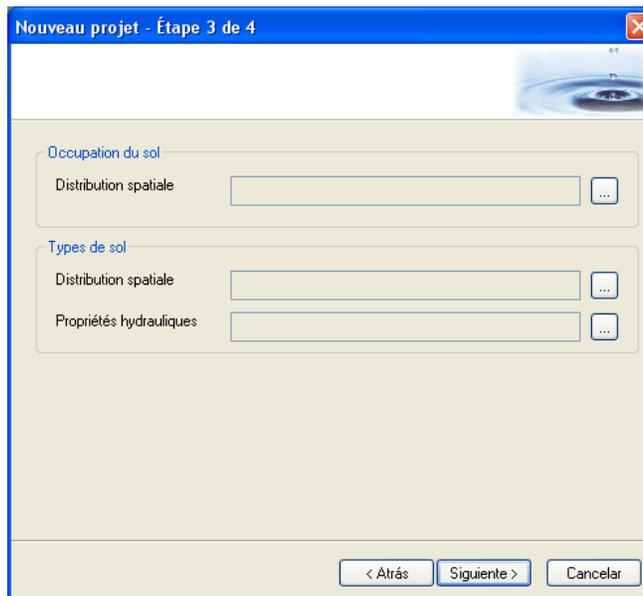


Figura 2.25 Interfaz de importación de archivos de uso y tipo de suelo de la cuenca para el montaje del proyecto en Hydrotel

Durante el proceso de importación de los archivos *Physitel* se crean carpetas en el subdirectorio donde se colocó el proyecto, las carpetas que se crean son *Physitel*, *Physio*, *Hydro*, *Meteo* y *HGM*, en donde estas últimas cuatro carpetas se encuentran vacías. A las carpetas *Physio*, *Hydro* y *Meteo* es necesario producirles sus archivos de alimentación al proyecto manualmente.

2.5.2 Definición de archivos fisiográficos

En la carpeta *Physio* es necesario agregar dos archivos en formato .txt, esto para que el modelo hidrológico cuente con la información del índice foliar y profundidad de raíces distribuidas en toda la cuenca. El índice foliar se encuentra contenido en el archivo LAI.txt (Fig. 2.26) y la profundidad de raíces en el archivo Raices.txt (Fig. 2.27).

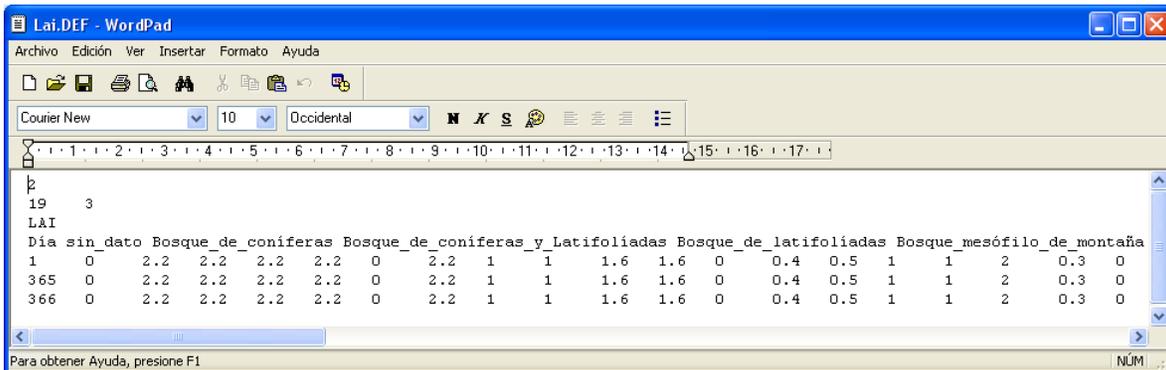


Figura 2.26 Archivo Lai.txt (índice foliar)

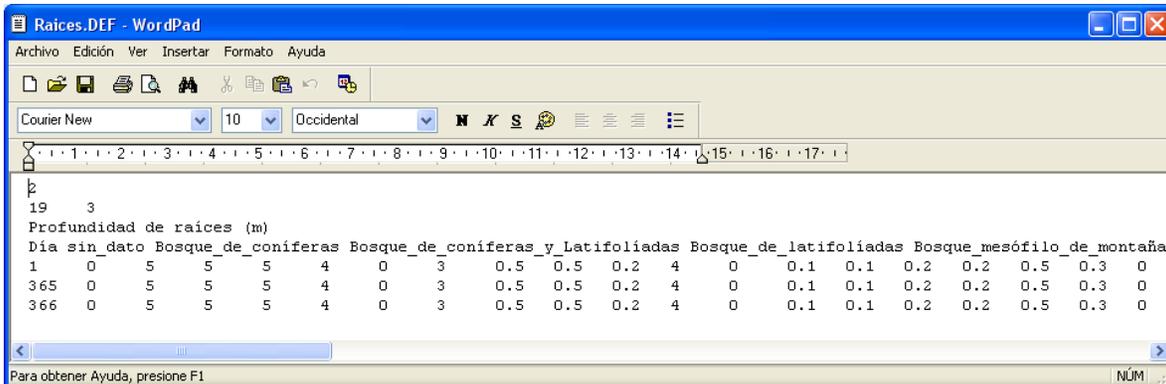


Figura 2.27 Archivo Raices.txt (profundidad de raíces)

2.5.3 Importación de archivos hidrométricos y meteorológicos

Para alimentar al modelo hidrológico es necesario crear archivos de control que contienen varios parámetros, de los cuales los más importantes son claves, coordenadas y altitud en el caso de las estaciones meteorológicas (Fig. 2.28 y 2.29). Estos archivos se colocan en la carpeta *Meteo* e *Hydro* respectivamente junto con sus series históricas.

meteo.stm - WordPad

Archivo Edición Ver Insertar Formato Ayuda

Courier New 10 Occidental

Estaciones climatológicas

09004MH	2798995.434	803690.142	2718	3	06	meteo\
09022MH	2796639.595	796066.814	3002	3	06	meteo\
09032MH	2812365.278	802581.527	2434	3	06	meteo\
09041MH	2801165.135	803027.933	2605	3	06	meteo\
09042MH	2805868.531	805331.360	2565	3	06	meteo\
09044MH	2811439.484	800844.536	2637	3	06	meteo\
09045MH	2814402.276	801335.622	2581	3	06	meteo\
09058MH	2812935.359	801795.840	2503	3	06	meteo\
15007MH	2838678.367	797629.209	2458	3	06	meteo\
15015MH	2839279.168	796845.120	2474	3	06	meteo\
15039MH	2827004.636	791450.098	2540	3	06	meteo\
15060MH	2829578.087	781811.071	2289	3	06	meteo\
15080MH	2839569.575	783966.026	2362	3	06	meteo\
15094MH	2828213.382	802888.334	2279	3	06	meteo\
15103MH	2842376.572	791423.744	2619	3	06	meteo\
15252MH	2838085.021	785098.545	2362	3	06	meteo\
15288MH	2841180.048	777282.730	2385	3	06	meteo\
17001MH	2822812.142	774760.774	1601	3	06	meteo\
17003MH	2820730.039	760078.547	1291	3	06	meteo\
17005MH	2820715.687	760753.412	1285	3	06	meteo\
17012MH	2818446.568	771323.214	1373	3	06	meteo\
17013MH	2805904.780	748905.937	1139	3	06	meteo\
17014MH	2791554.207	765040.835	1284	3	06	meteo\
17017MH	2822422.916	767019.571	1407	3	06	meteo\
17018MH	2802921.426	754738.617	960	3	06	meteo\
17024MH	2813165.120	765474.192	1367	3	06	meteo\

Para obtener Ayuda, presione F1

Figura 2.28 Archivos de control de las estaciones seleccionadas meteorológicas

hydro.sth - WordPad

Archivo Edición Ver Insertar Formato Ayuda

Courier New 10 Occidental

commentaire

0018193	2810092.33	769982.96	0	3	05	hydro
0018223	2804631.62	758363.80	0	3	05	hydro
0018302	2832660.72	779669.66	0	3	05	hydro

Para obtener Ayuda, presione F1

Figura 2.29 Archivos de control de las estaciones seleccionadas hidrométricas

Las series históricas meteorológicas contienen información diaria de precipitación, temperatura máxima y temperatura mínima, de las cuales el modelo hidrológico genera un escurrimiento sobre la cuenca. Las series históricas hidrométricas contienen información diaria de caudales que sirven de comparación para realizar la calibración del modelo. Todo esto para cada una de las estaciones meteorológicas e hidrométricas contenidas en sus respectivos archivos de control. En la figura 2.30 y 2.31 se muestra la ubicación de las estaciones meteorológicas e hidrométricas respectivamente en plataforma *Hydrotel*.

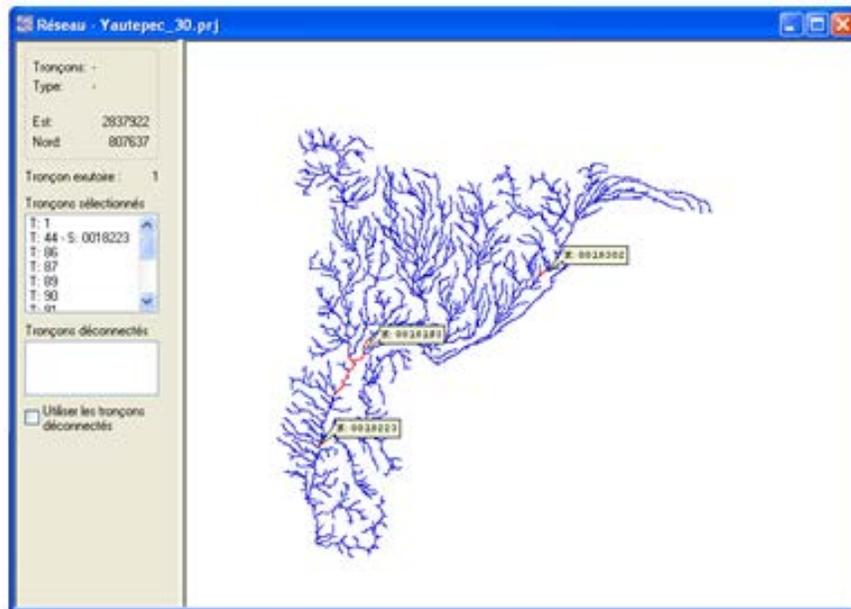


Figura 2.30 Localización de estaciones hidrométricas

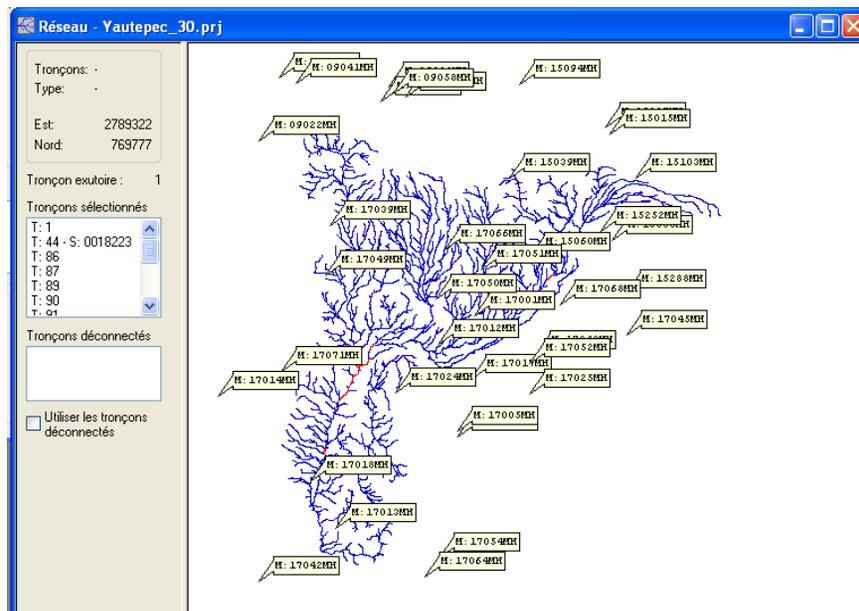


Figura 2.31 Localización de estaciones meteorológicas

2.5.4 Creación de grupos de UHRH

Para calibrar el modelo hidrológico es necesario crear grupos, los cuales deben contar con una estación de calibración situada a la salida de cada grupo (subcuenca), para así en cada uno de ellos modificar algunos parámetros fisiográficos e hidráulicos. Se integraron dos grupos formados dentro del modelo hidrológico (Fig. 2.32).

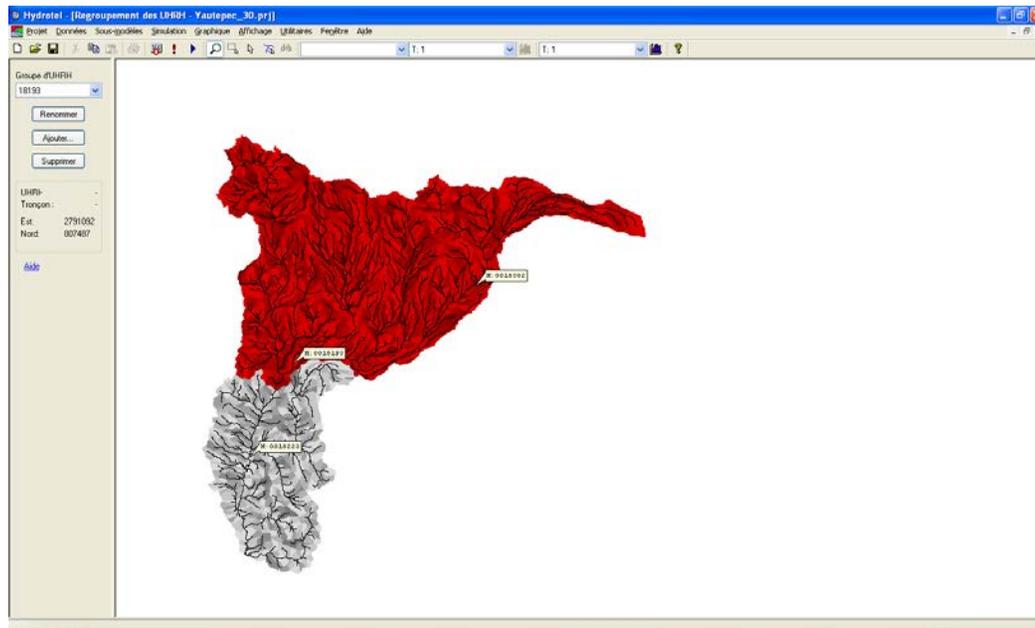


Tabla 2.32 Grupos formados en la cuenca

2.5.5 Definición de gradientes temperatura

Una de las funciones de los modelos de simulación de *Hydrotel* es el de interpolar los datos medidos en las tres estaciones meteorológicas más cercanas a la UHRH (precipitación, temperatura máxima y temperatura mínima), para obtener una distribución espacial de esas variables.

Cuando se interpolan las temperaturas del aire en cada estación meteorológica sobre las UHRH se toma en cuenta la variación espacial de los valores, pero también es importante definir la variación espacial con respecto a la altitud entre la estación y cada UHRH. *Hydrotel* corrige las temperaturas por cada 100 m de altura en cada grupo, la ecuación de corrección es la mostrada en el inciso 1.5.5.

Para obtener los gradientes de temperatura de cada grupo (Tabla 2.1) primero se determinaron las temperaturas promedio anuales de cada estación, estas se agruparon conforme a los grupos

de calibración para así calcular el gradiente vertical de temperatura. En la interfaz de aplicación de *Hydrotel* se introducen cada uno de los valores calculados (Fig. 2.33).

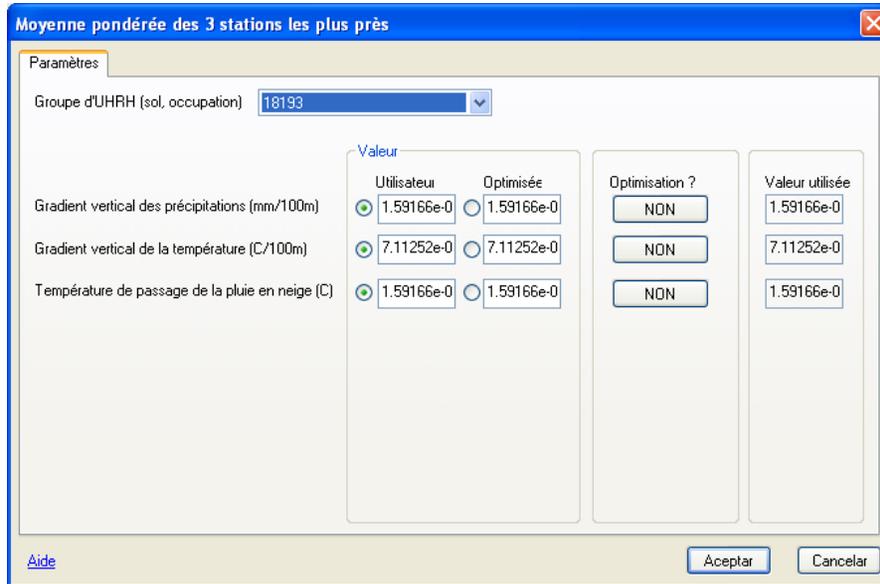


Figura 2.33 Interfaz de gradientes verticales de temperatura

En la tabla 3.1 se muestran los gradientes de temperatura que se obtuvieron para cada grupo.

Grupos	18193
Gradiente de temperatura	-0.63

Tabla 2.1 Valores de gradientes verticales de temperatura

2.5.6 Generación de hidrogramas geomorfológicos

Una vez creado el proyecto *Hydrotel* para la modelación hidrológica de la cuenca se procede a simular el modelo en toda el área de proyecto. Esta simulación se realiza para crear los archivos de los hidrogramas geomorfológicos (HGM) dentro de la carpeta *HGM*.

Los archivos *HGM* contienen los hidrogramas geomorfológicos de cada UHRH (Fig. 2.34), es decir, hidrogramas que describen la capacidad de respuesta del escurrimiento en cada UHRH. Los HGM dependen del coeficiente de *Manning* y de la pendiente de los UHRH.

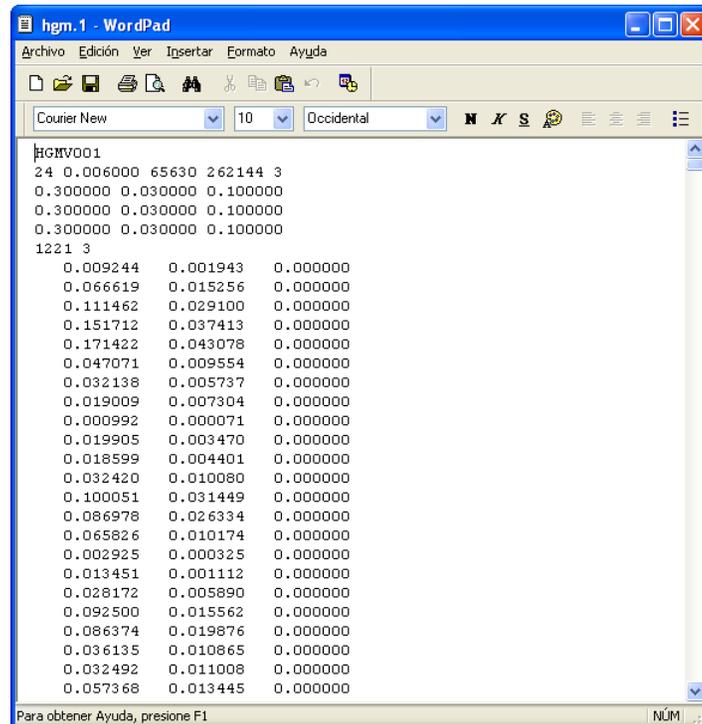


Figura 2.34 Archivo de datos del hidrograma geomorfológico

Los archivos *HGM* se generan una sola vez al iniciar la primera simulación para toda la etapa de proyecto al menos que haya un cambio en la conformación de los grupos o coeficiente de *Manning*.

2.5.7 Calibración del modelo hidrológico

Una vez concluida la etapa de generación de archivos *HGM*, como se mencionó en el inciso anterior, se procede a la calibración de grupos siguiendo el procedimiento que se enumera a continuación:

4. Elección del tramo de la red hidrográfica que funcionará como salida de la subcuenca, es decir, del grupo que será calibrado.
5. Elección de submodelos. Estos son los algoritmos alternativos vistos en el inciso 2.2.
6. Simulación de calibración. La simulación se realiza con un periodo de tiempo de siete años y como condiciones iniciales se ajustan las humedades de las capas 1, 2 y 3.

En esta etapa los parámetros de ajuste son:

- *Humedades de las capas 1, 2 y 3 para las condiciones iniciales*
- *Factor de evapotranspiración*
- *Profundidades de las capas 1, 2 y 3*
- *Coefficiente de recesión*

La calibración de cada grupo se realiza con procedimientos manuales iterativos, apoyados con un esquema de optimización de parámetros, considerando el ajuste en volumen, forma y sincronización del hidrograma.

Del mismo modo, el ajuste en volumen se realizó obteniendo el mínimo porcentaje del error relativo de la diferencia entre el volumen observado (dado por los registros de las series hidrométricas) y el volumen simulado. La forma y sincronización del hidrograma se llevó a cabo considerando el mayor valor de un índice estadístico, este índice es el coeficiente llamado *Nash Sutcliffe* que considera el caudal simulado, el caudal observado y el caudal medio en el periodo simulado, las ecuaciones se muestran en el inciso 1.5.7.

En la tabla 2.2 se muestran los valores de calibración de los grupos de la cuenca, el índice de error estadístico *Nash Sutcliffe* y el error relativo de la diferencia entre volúmenes.

Grupo	18193
Subcuenca	Yautepec
Periodo	1995
Factor ETP	2.25375
Espesor ₁	0.1
Espesor ₂	0.2
Espesor ₃	0.5
CR	1.73E-10
$\Delta V(\%)$	76.8
NS	-0.72

Tabla 2.2 Valores de calibración de grupos

En la figura 2.35a, se muestra la subcuenca que se utilizaron para la calibración del grupo 18193, y en la figura 2.35b, la comparación de los hidrogramas observados con los hidrogramas simulados.

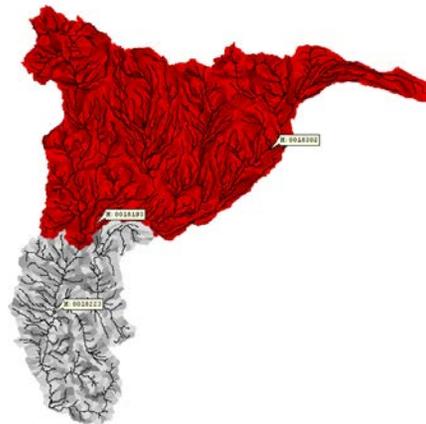


Figura 2.35a Calibración del grupo 18193, subcuenca Yautepec

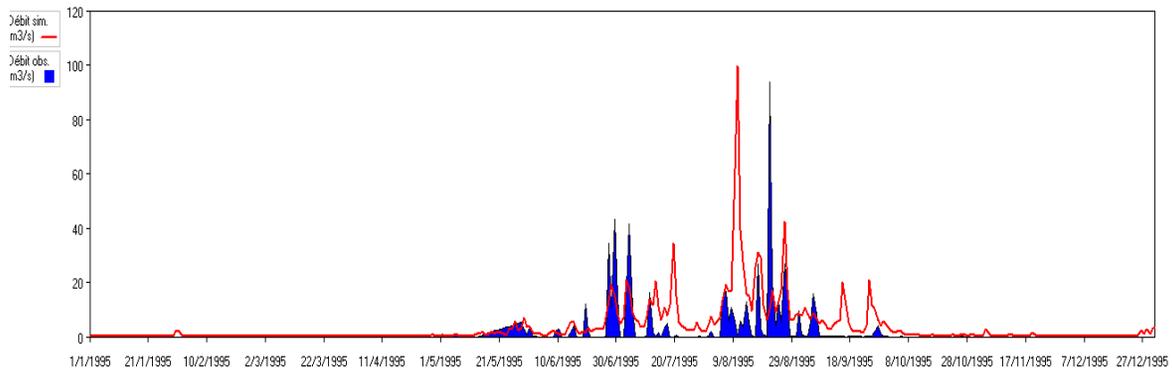


Figura 2.35b Hidrograma de calibración del grupo 18193, subcuenca Yautepec

2.5.8 Simulación de tramos en confluencia del río Yautepec

Calibrado el modelo hidrológico, se realizó una simulación de siete años, del año 1995 al año 2001. En la figura 2.36 a y b se muestran los escurrimientos diarios de la simulación de los siete años.

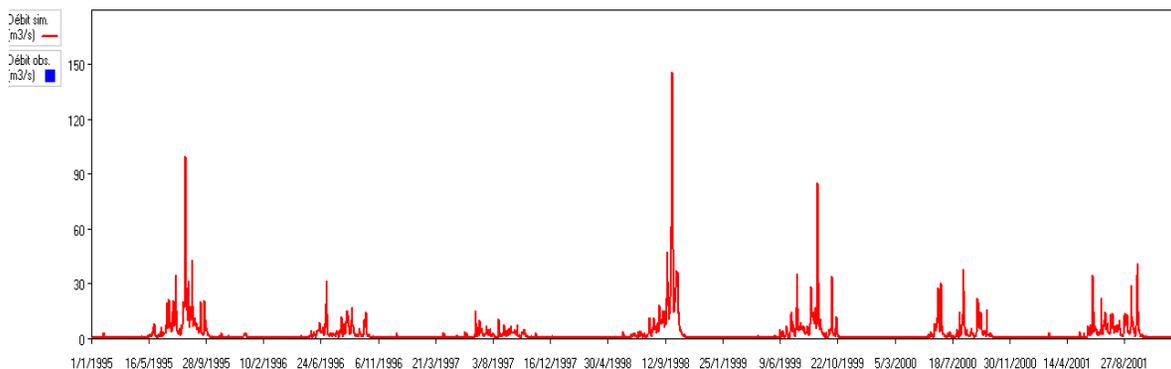


Figura 2.36a Hidrograma de simulación aguas abajo de la estación Yautepec

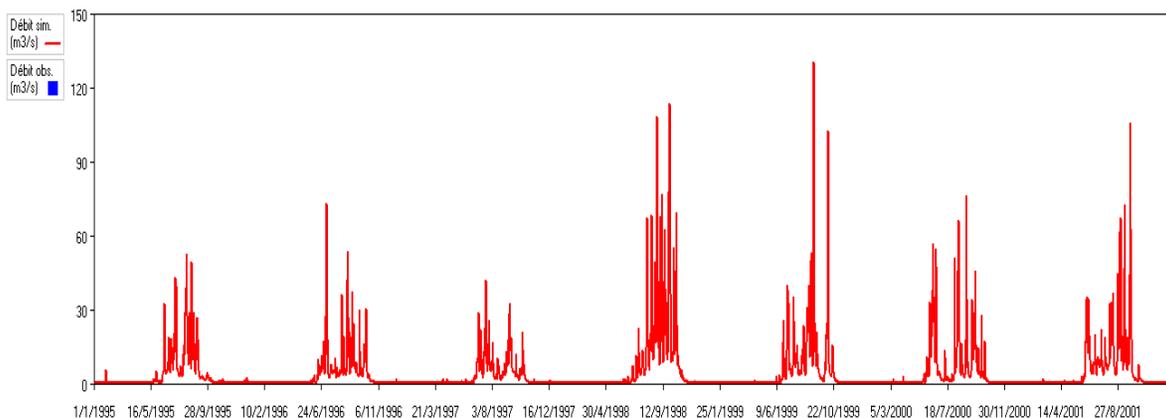


Figura 2.36b Hidrograma de simulación en el tramo confluyente al río Yautepec

2.6 Proyecto HEC-RAS

En lo que respecta al software para llevar a cabo la modelación hidráulica de inundación, se emplearon: el HEC-RAS 4.1.0 (*United States Army Corps, 2012*) y ArcGIS 9.3 (ESRI, 2012). El tramo en estudio comprende aproximadamente 6 km, el cual inicia en la estación hidrométrica “Yautepec” localizada en la margen derecha del Puente “La Perseverancia” y concluye 1 km aguas debajo del cruce del río Yautepec con la Carretera Federal a Cuautla (Fig. 2.37). A la mitad del tramo en estudio, se incorpora como tributario el río Apanquetzalco. Las márgenes presentan diferentes usos de suelo, que van desde zonas agrícolas hasta zonas completamente urbanas.

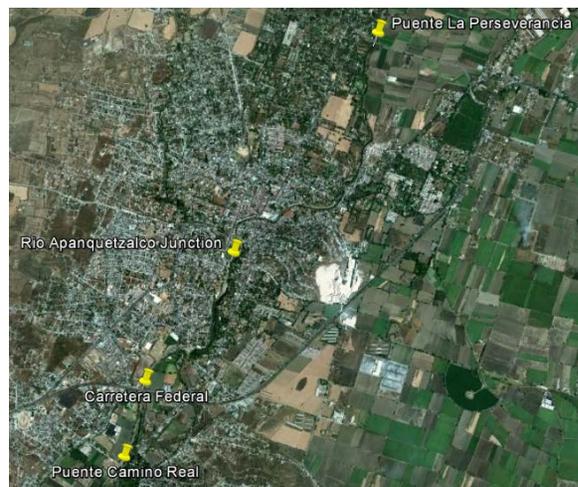


Figura 2.37 Tramo en estudio (imagen cortesía de Google Earth)

2.6.1 Recopilación y análisis de información

Uno de los primeros pasos fue la obtención de elevaciones reales del terreno natural en campo utilizando sistemas de posicionamiento global (GPS por sus siglas en inglés) figuras 2.38a y b, así como la obtención de las dimensiones geométricas de las estructuras hidráulicas (puentes) que afectan la planicie de inundación, dado que restringen el flujo y se genera un efecto de remanso aguas arriba de la estructura. Para tal efecto, se utilizó un equipo *Trimble Survey Controller* para la obtención de puntos geodésicos.



Figura 2.38a Equipo GPS



Figura 2.38b Trabajo de campo: obtención de puntos geodésicos

2.6.2 Preparación de datos

Tomando como base las cartas vectoriales de INEGI escala 1:50,000 y los puntos geodésicos obtenidos, se generó una superficie *TIN* (Triangular Irregular Network) utilizando la extensión de *Spatial Analyst* de ArcGIS. Posteriormente se trazaron 18 secciones transversales, las cuales se presentaron un promedio de separación de 335 metros. El diseño de las mismas se realizó trazándolas perpendiculares a las curvas de nivel y al eje central de la corriente (Fig. 2.39).

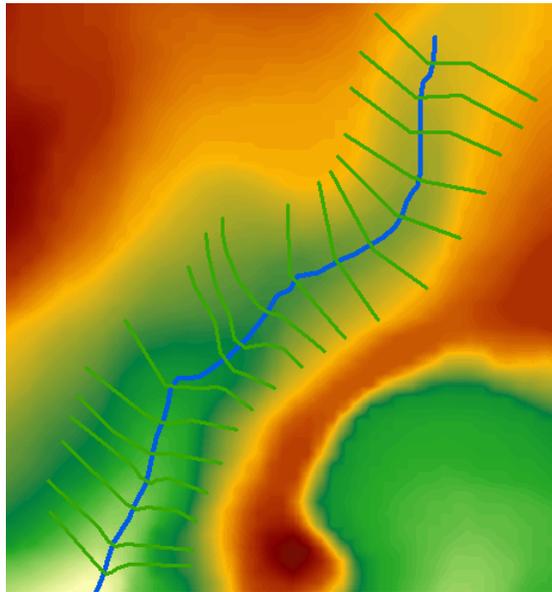


Figura 2.39 TIN TN y secciones transversales

2.6.3 Modelo HEC-RAS

El HEC-RAS es un modelo unidimensional de régimen permanente, gradualmente variado, que genera como resultado el perfil de la superficie libre de agua calculando el nivel, velocidad de agua y área mojada en cada tramo o sección transversal del río.

El eje del río y las secciones transversales se exportaron a HEC-RAS utilizando la extensión 3D Analyst de ArcGIS, utilizando parámetros convencionales para las coeficientes de pérdida (rugosidad, expansión y contracción). Durante el 21 de Agosto de 1995 se presentó un evento extremo, de acuerdo al registro de la estación hidrométrica Yautepec, el gasto medio horario fue de $231 \text{ m}^3/\text{s}$ (8 pm), hasta dicho punto el área de drenaje es de 527.81 km^2 . Mientras que el área de drenaje del río Apanquetzalco al unirse con el río Yautepec es de 247.50 km^2 y asumiendo una relación lineal, se estima que el caudal de aporte del tributario fue de aproximadamente

108 m³/s durante el evento mencionado. Lo anterior representa un gasto total de 339 m³/s a partir de la sección transversal (*River Station, RS*) 92 en el modelo.

2.6.4 Mapeo de la planicie de inundación

Utilizando la extensión 3D Analyst de ArcGIS, se generó un TIN WS con las elevaciones resultantes del modelo HEC-RAS en cada sección transversal. Finalmente, para obtener la planicie de inundación se calcula la diferencia espacial entre los dos TINs, esta operación genera tres posibilidades, siendo la última la planicie de inundación:

$$TIN_{TN} - TIN_{WS} = \begin{cases} 1 \text{ (Above)} \\ 0 \text{ (Same)} \\ -1 \text{ (Below)} \end{cases} \rightarrow \text{Planicie de inundación}$$

La figura 2.40 muestra la planicie de inundación y las elevaciones del agua para cada una de las secciones transversales.

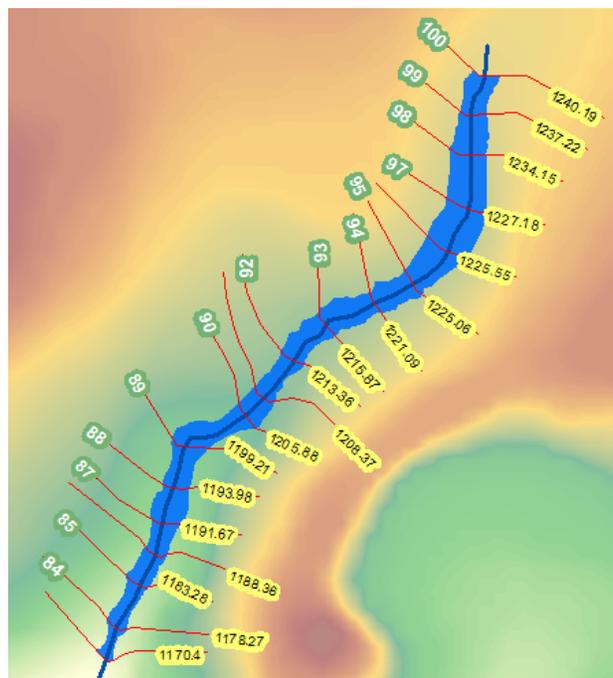


Figura 2.40 Planicie de inundación (Agosto 21, 1995)

3. Interfaz de usuario.

La interfaz de usuario se desarrolló una aplicación de escritorio en el lenguaje de programación C#.net Framework 3.5, la cual consiste en comunicar los modelos hidrológicos *Hydrotel* y un sistema que modela el flujo de agua en cauces (HEC-RAS). Esto con el fin de realizar una simulación completa que inicia con un pronóstico de lluvia y termina con la delimitación del área inundable, de tal manera que el usuario pudiera de forma sencilla, rápida y eficiente determinar zonas de riesgo para pronosticar de forma más confiable eventos de inundación en un área en específico.

Debido a los requerimientos en el desarrollo de este proyecto se realizó la implementación de una metodología de desarrollo de software iterativa, que permite obtener resultados al final de cada iteración. Por lo que se aplicó una adaptación del proceso unificado de Desarrollo de Software (RUP) ya que provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

Una iteración completa de esta metodología se compone de las fases de administración y análisis de requerimientos y de análisis y diseño (Fig. 3.1).

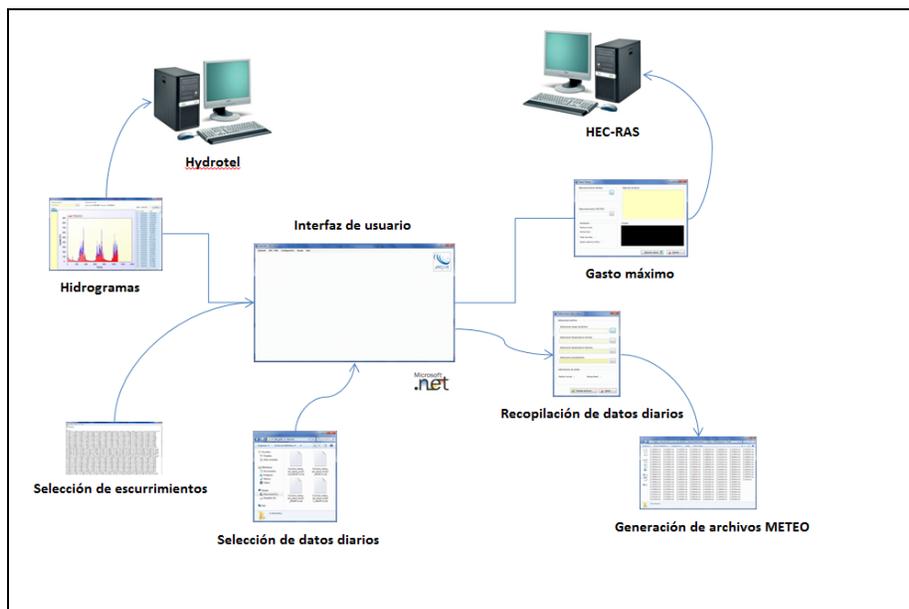


Figura 3.1 Mapa conceptual de la aplicación

Administración de Requerimientos. La principal dificultad con los esfuerzos de ingeniería de procesos, es que la ingeniería de software y el grupo de trabajo que ejecuta los procesos actuales no se comunican adecuadamente entre sí. Esto origina que los resultados de la ingeniería de procesos no sean usados adecuadamente como entrada para los esfuerzos de desarrollo de software. Esta etapa permitió mostrar, cómo crear y mantener directamente la rastreabilidad entre los procesos y/o mecanismos que se llevan actualmente de manera manual para recopilar e integrar la información y los procesos optimizados.

En esta etapa se describió lo que el sistema debiera hacer y permitió identificar la problemática que se generaría al comunicar dichas plataformas, asimismo la descripción y el funcionamiento del sistema.

Análisis de requerimientos. Se identificaron las necesidades de los posibles usuarios con el sistema (interfaz de usuario), así como también a los requerimientos de los modelos hidrológicos e hidráulico con los se interactuó cuando la solución fue implementada. Se realizaron pruebas para representar el comportamiento del sistema; obteniendo así los requerimientos funcionales, no funcionales y las restricciones al diseño e implementación.

Análisis y Diseño. Esta etapa fue conformada por los procesos de diseño general del sistema, desarrollo e implementación. En la etapa de diseño general del sistema, se definió la arquitectura del sistema, mediante los requerimientos de diseño y desarrollo, así como la especificación del entorno tecnológico. Se consideraron los aspectos de compatibilidad, tolerancia a fallos, modularidad, empaquetado y distribución. De acuerdo al diseño establecido se implementaron en paralelo los módulos que componen al sistema.

3.1 Descripción de ventanas del mapa conceptual

La metodología seguida para abordar la integración de las dos plataformas de simulación fue en primera instancia la modelación, calibración y validación de la hidrología e hidráulica del fenómeno.

Para lograr esto, fue necesario recopilar, analizar e interpretar los datos de entrada y de salida que generaban los modelos *Hydrotel* y *HEC-RAS*. El resultado del modelo hidrológico es insumo del modelo hidráulico, y este proceso es administrado dentro de la interfaz de usuario. Dentro de la interfaz de usuario existe una barra de menú que contiene a su vez, un conjunto de submenús que tienen relación con el modelo hidrológico *Hydrotel* y, el modelo hidráulico *HEC-RAS*, la configuración de las rutas de cada uno de los modelos y una ayuda al usuario final del sistema (Fig. 3.2).

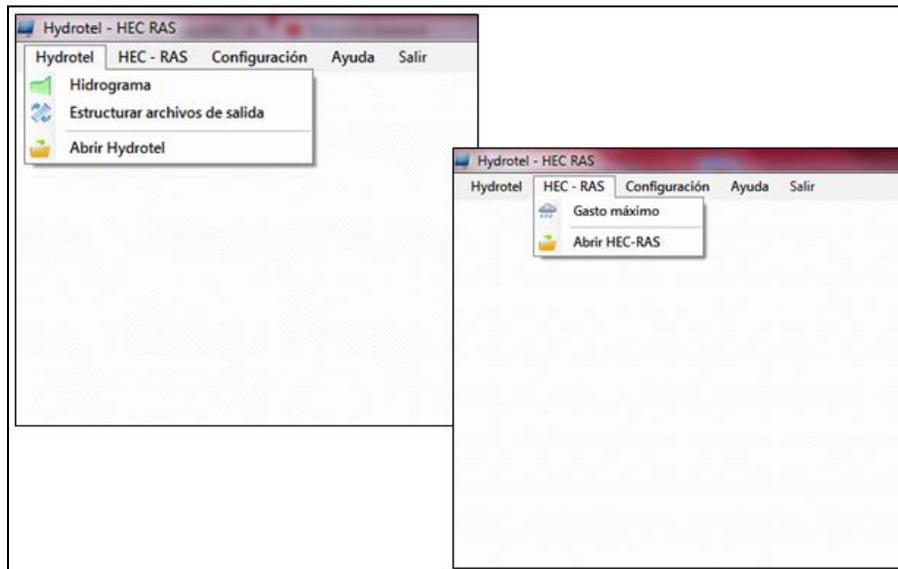


Figura 3.2 Estructura de la interfaz de usuario

3.2 Modelo hidrológico *Hydrotel*

La interfaz contiene un conjunto de opciones que ofrecen al usuario el manejo de archivos de entrada y salida que el modelo hidrológico *Hydrotel* necesita cuando se ejecuta la opción *Abrir Hydrotel*, automáticamente se desprenderán una serie de pantallas indicando al usuario que es lo que necesita realizar dentro del programa *Hydrotel* para llevar a cabo la simulación.

Con base en la modelación hidrológica de la cuenca del río Yautepec así como su calibración y validación, descrita en capítulos anteriores, de obtuvieron dos hidrogramas de salida en dos puntos de la cuenca, en el punto denominado La perseverancia que es donde inicia la zona urbana de la localidad de Yautepec y el punto de confluencia del río Yautepec y el río Apanquetzalco.

La simulación modelo hidrológico puede activarse dentro de la interfaz de usuario en la cual se le indica la ruta de los insumos (series de lluvia, temperatura máxima y mínima), el resultado de la simulación es un hidrograma mismo que será insumo del modelo hidráulico (Fig. 3.3a,b,y c).

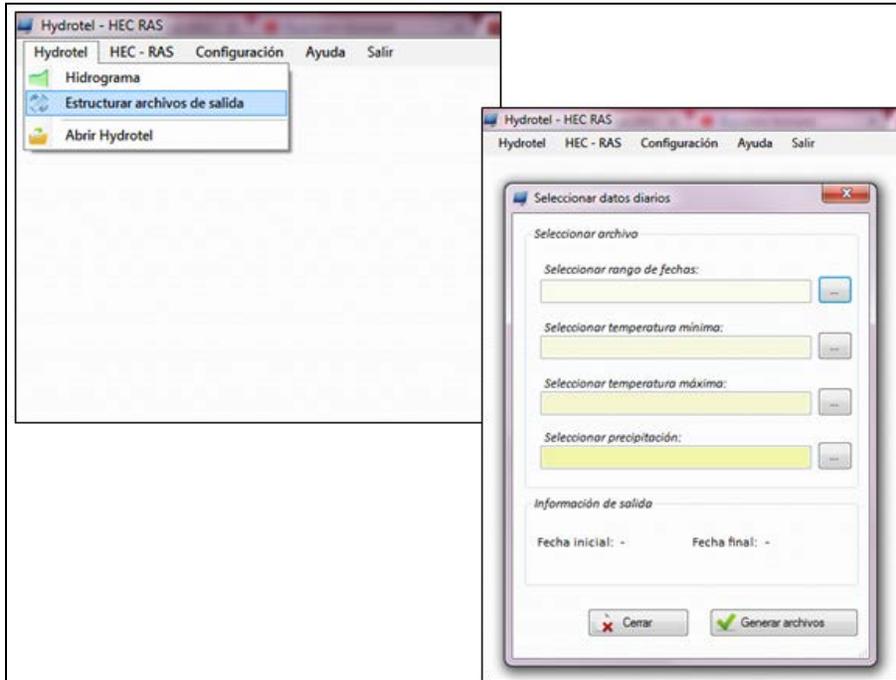


Figura 3.3a Requerimientos del modelo hidrológico
(Precipitación, Temperatura máxima y mínima)

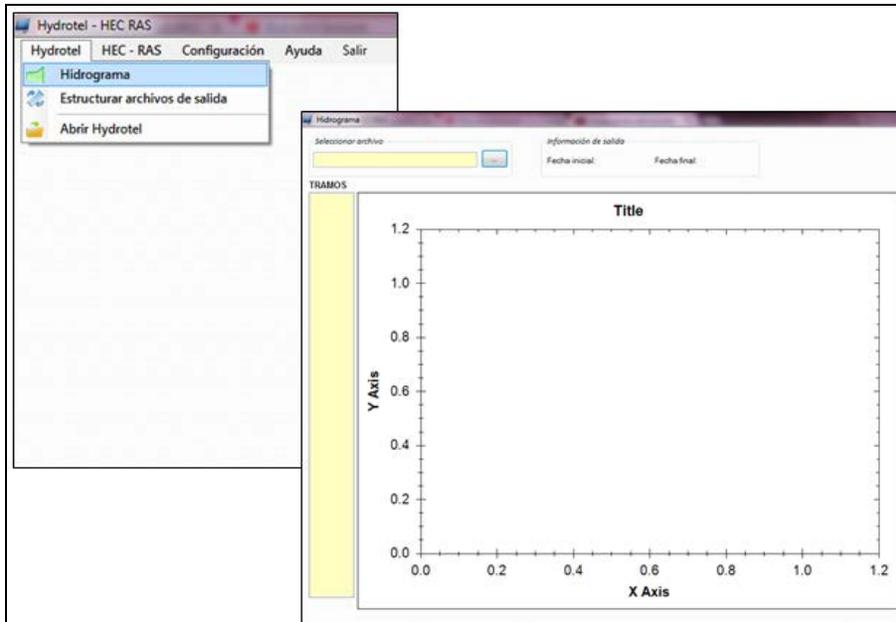


Figura 3.3b Hidrograma de salida del modelo hidrológico

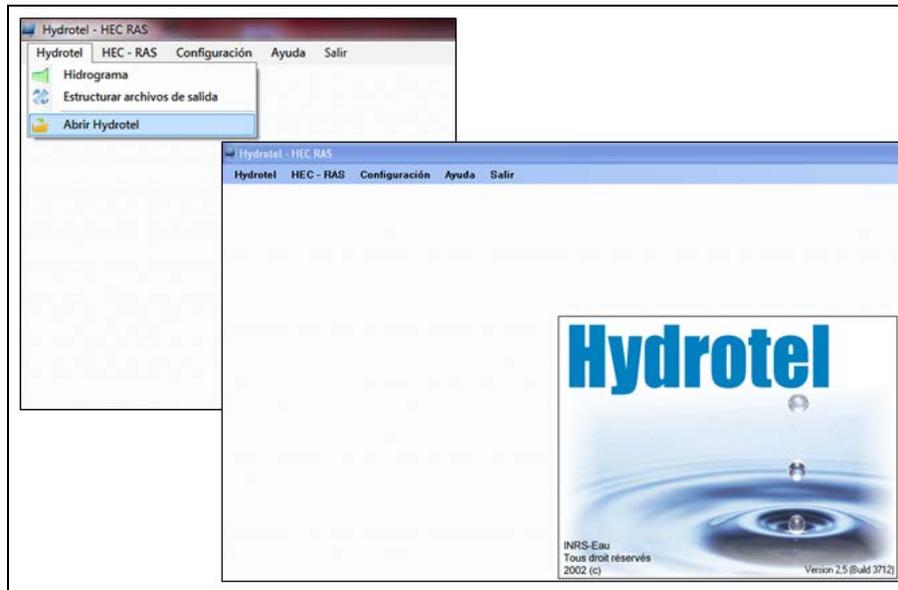


Figura 3.3c Operación del modelo hidrológico

Para generar un hidrograma que represente los diversos gastos a partir de un determinado periodo de tiempo mediante el análisis de un periodo en específico, es necesario alimentar el sistema con un archivo con extensión *.txt que contenga datos acerca de precipitación, dicho archivo debe ser un archivo de salida generado por *Hydrotel*, el cual contiene información del caudal a lo largo del periodo de tiempo del estudio.

Para mostrar un hidrograma en pantalla es necesario seleccionar que el usuario examine y cargue el archivo, para realizar esta acción el usuario realiza un evento clic sobre el botón que se encuentra en la sección *Seleccionar archivo* que se encuentra dentro de la pantalla Hidrograma (fig. 3.4). El sistema automáticamente valida el archivo que contenga una estructura de *Hydrotel* para que sea leído correctamente. Si el archivo es correcto, se cargan los tramos encontrados en el archivo dentro de una lista en la sección *tramos* y envía un mensaje al usuario indicando que debe seleccionar el tramo que desea ver en pantalla, una vez seleccionado el tramo el sistema devuelve un hidrograma con los datos de precipitación contenidos en el archivo. En la sección *Información de salida* se observa el rango de fechas que contiene el archivo que ha sido seleccionado.

Una vez representados los datos de precipitación en el hidrograma, el sistema proporciona un componente para la selección de una fecha, en caso que el usuario desee encontrar un punto en específico basta con seleccionar la fecha y realizar clic con el botón derecho del mouse en el botón Buscar. Se puede apreciar mejor el flujo de información mediante el siguiente algoritmo:

- El usuario inicia la aplicación, y el sistema espera el archivo de simulación de *Hydrotel* como archivo de entrada.
- El sistema lee los datos de entrada del archivo de simulación generado por *Hydrotel*.
- El sistema determina el rango de *debits* e indica al usuario por medio de un componente *ListBox*, los tramos que puede seleccionar para ser graficados
- El usuario selecciona el tramo de interés.
- El sistema lee e interpreta el tramo seleccionado, y determina la entrada de '*Debits*' correspondiente.
- El sistema lee la información del gasto y empieza a graficar por medio de una librería de C# de gráficos (*ZedGraph*).
- El sistema termina de graficar y presenta en pantalla los resultados de manera gráfica (Fig3.4).

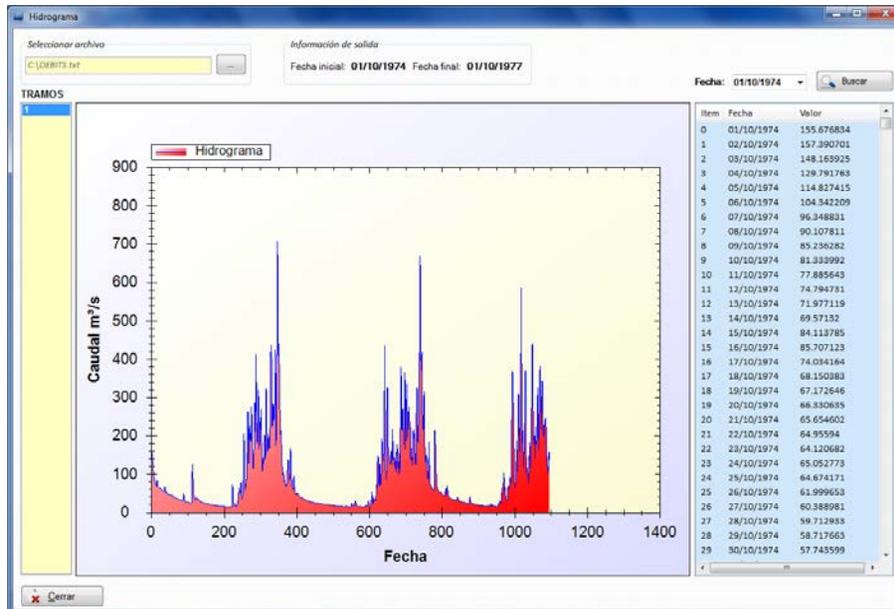
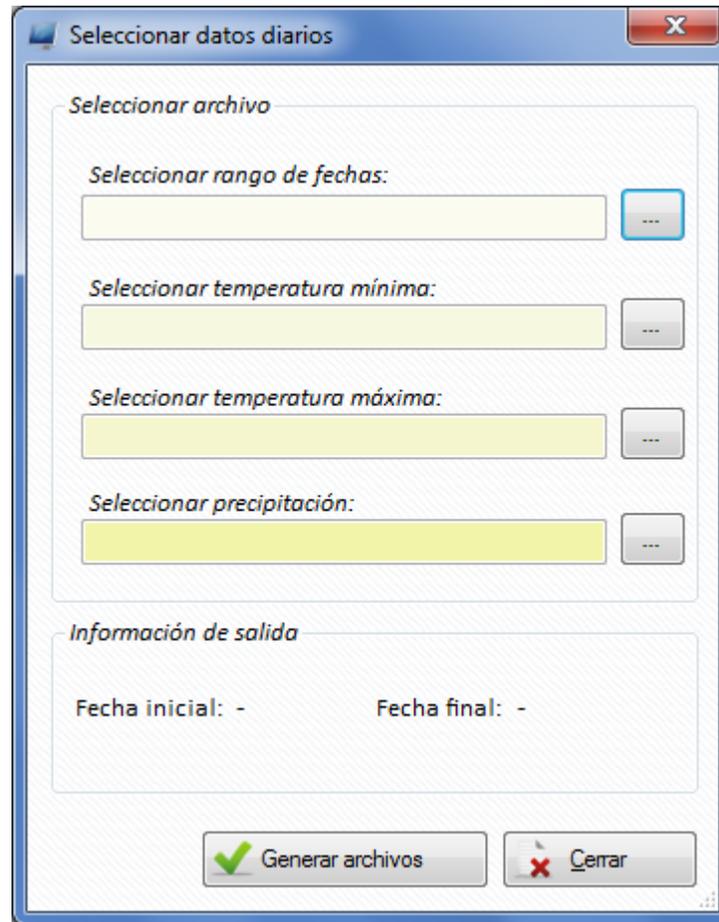


Figura 3.4 Pantalla hidrograma

Para generar los archivos con extensión **.met*, que serán insumos por la simulación en *Hydrotel*, son necesarios archivos con extensión **.txt* que contenga datos diarios de temperatura mínima, máxima, precipitación y una serie de fechas, estos cuatro archivos son requeridos en la pantalla de *Estructurar archivos de salida* (Fig. 3.5) y son seleccionados con su respectivo botón con el fin de interpretar y estructurar estos datos de tal forma que su sintaxis sea en su totalidad compatible en la estructura en que son leídos en el modelo *Hydrotel*.



Seleccionar datos diarios

Seleccionar archivo

Seleccionar rango de fechas:

Seleccionar temperatura mínima:

Seleccionar temperatura máxima:

Seleccionar precipitación:

Información de salida

Fecha inicial: - Fecha final: -

Generar archivos Cerrar

Figura 3.5 Pantalla Estructurar archivos de salida

Una vez ingresados estos archivos el usuario realiza el evento (clic en el botón Generar archivos), el sistema valida que los archivos contengan datos numéricos y que sean archivos válidos, si es así, comienza a interpretar y transformar estos datos en una estructura compatible a *Hydrotel* generando dentro de una carpeta llamada *Meteo* archivos con la nueva estructura. Se puede apreciar mejor el flujo de información mediante el siguiente algoritmo:

- El usuario inicia la aplicación, y el sistema espera los archivos de fechas, temperatura máxima, mínima así como su respectiva precipitación.
- El sistema lee los datos de entrada de los archivos.
- El sistema construye y relaciona los datos de los archivos para generar una salida compatible para *Hydrotel*
- El sistema genera un conjunto de archivos con extensión **.met* dentro de una carpeta llamada *Meteo*.

La acción que debe realiza el usuario es, clic con el botón derecho del mouse sobre la opción “Abrir Hydrotel”, el sistema automáticamente ejecuta dicho programa sobre la aplicación siempre y cuando éste se haya asignado, por lo menos una vez, la ruta en el campo *Hydrotel* en la pantalla de *configuración del sistema* (Fig. 3.11) y la aplicación *Hydrotel* haya sido instalada en la computadora.

En caso de que la ruta no sea correcta o el programa no haya sido instalado en la computadora, el sistema envía un mensaje de error (Fig. 3.6).

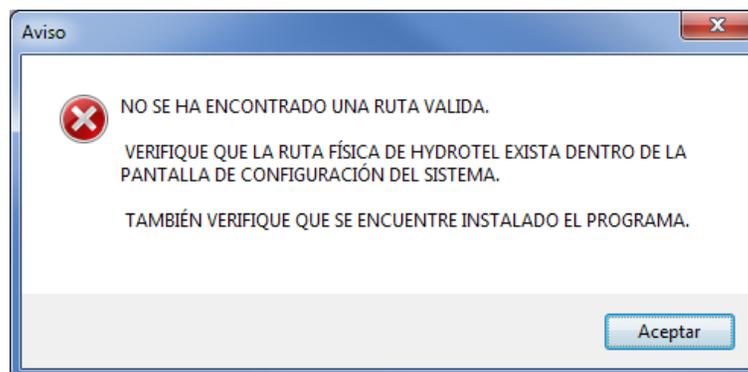


Figura 3.6 Mensaje de error al no encontrar la aplicación Hydrotel

Si la dirección es correcta y el evento clic sobre el botón “Abrir Hydrotel” es realizado, la aplicación ejecuta la aplicación (fig. 3.7).

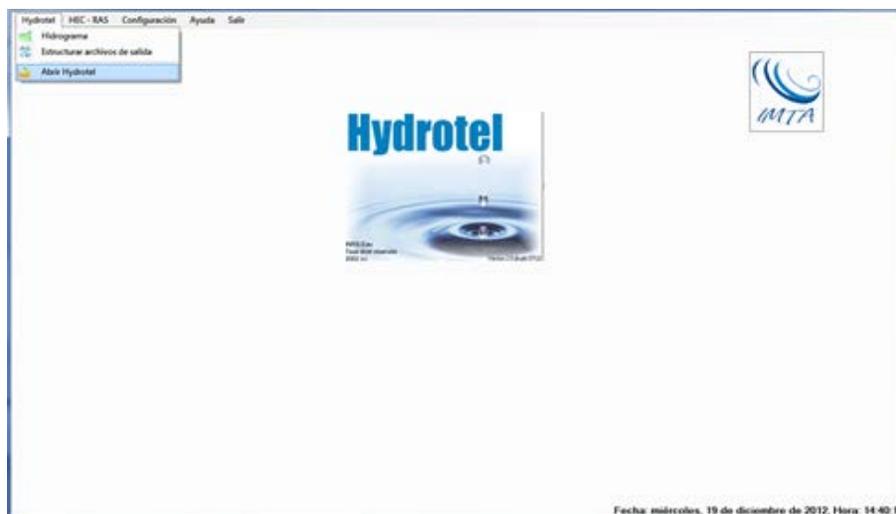


Figura 3.7 Ejecución de Hydrotel

3.3 Modelo hidráulico *HEC-RAS*

El modelo hidráulico se realizó en un tramo del río Yautepec de aproximadamente seis kilómetros que cubre la parte donde se tienen mayores afectaciones ante precipitaciones extraordinarias. Este modelo toma uno o varios hidrogramas de salida del modelo hidrológico, en el presente análisis la modelación hidráulica toma dos hidrogramas, el primero al iniciar al tramo de simulación y el segundo se integra en un punto intermedio del tramo en estudio.

La acción que debe realiza el usuario es, clic con el botón derecho del mouse sobre la opción "*Abrir HEC-RAS*", el sistema automáticamente ejecuta dicho programa sobre la aplicación siempre y cuando éste se haya ingresado por lo menos una vez la ruta en el campo HEC-RAS en la pantalla de configuración del sistema (Fig. 3.11) y la aplicación HEC-RAS haya sido instalada en la computadora.

Si la dirección es correcta y el evento clic sobre el botón "*Abrir HEC-RAS*" es realizado, la aplicación ejecuta la aplicación (fig. 3.8).

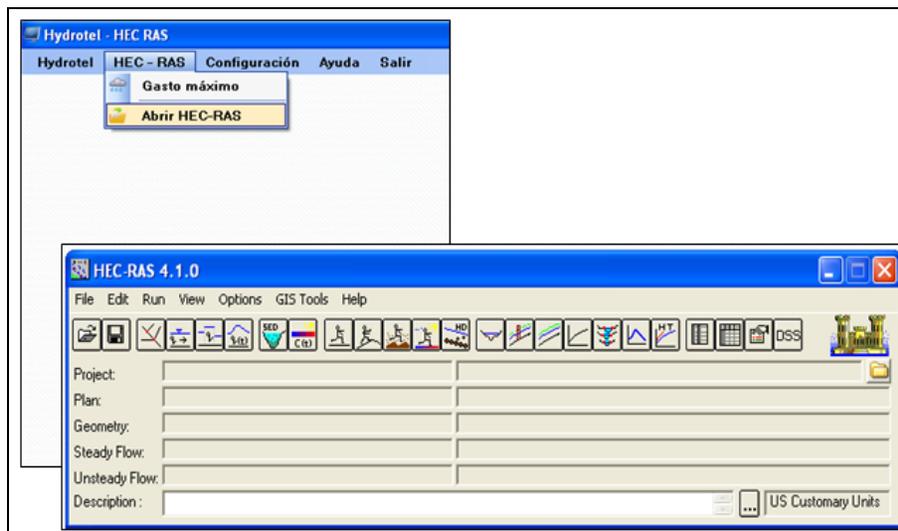


Figura 3.8 Ejecución de HEC-RAS

En caso de que la ruta no sea correcta o el programa no haya sido instalado en la computadora, el sistema envía un mensaje de error (Fig. 3.9).

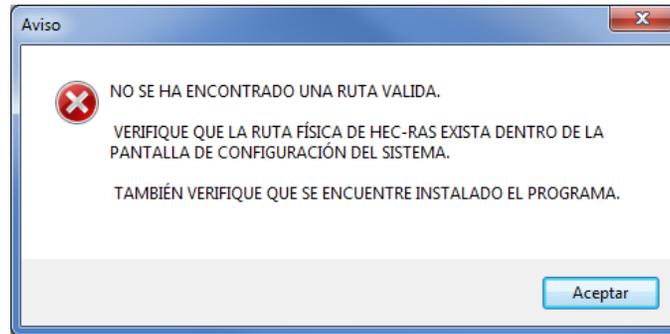


Figura 3.9 Mensaje de error al no encontrar la aplicación HEC-RAS

La interfaz cuenta con una función donde puede seleccionarse el tramo de río de interés y determinar el hidrograma y el gasto máximo.

Para generar el archivo que será insumo del modelo *HEC-RAS*, es necesario que el sistema en primer lugar lea un archivo de texto con extensión *.txt, éste archivo debe de contener una serie de datos que correspondan a precipitación, una vez ingresado al sistema este archivo, se ejecuta un conjunto de instrucciones, es decir, la interfaz interpreta los datos y crea como salida un archivo nuevo que servirá como entrada para el modelo hidráulico, de tal forma que su sintaxis sea en su totalidad compatible en el lenguaje *HEC-RAS*.

Para empezar a generar el cálculo máximo sobre un archivo de simulación, es necesario que el usuario examine y cargue el archivo, para realizar esta acción el usuario realiza un clic sobre el botón que se encuentra en la pantalla de Gasto máximo (figura 3.2) en la sección “*Seleccionar archivo de flujos*”. Una vez seleccionado el archivo, el usuario debe dar clic en un elemento de la lista que contiene todos los tramos, el primer gasto seleccionado, se considera como el gasto principal y los tramos seleccionados posteriormente, se interpretan como gastos tributarios, estas acciones se realizan en la sección “*Selección de tramos*” de la figura mencionada con anterioridad.

Al término del proceso de selección del gasto principal y los gastos tributarios, el usuario tendrá que seleccionar el archivo de flujo de *HEC-RAS* para vaciar la información calculada y seleccionada por el usuario, realizando en este proceso el cálculo de los gastos máximos de cada tramo seleccionado (gasto principal y/o gastos tributarios), esta acción se realiza en la sección *Seleccionar archivo HEC-RAS*, sobre el botón en la figura 3.2.

Las acciones que realiza el usuario se observan en la bitácora de las actividades del sistema, (sección *consola* de la figura 3.2), la cual contendrá las líneas de cada acción realizada. Se puede apreciar mejor el flujo de información mediante el siguiente algoritmo:

- El usuario inicia la aplicación, y el sistema espera el archivo de simulación de *Hydrotel* de entrada.

- El sistema lee los datos de entrada del archivo de simulación generado por *Hydrotel*
- El sistema determina el rango de 'DEBITS' e indica al usuario por medio de un *ListBox* los tramos que puede seleccionar para ser graficados
- El usuario selecciona el tramo de interés
- El sistema lee e interpreta el tramo seleccionado, y determina la entrada de *Debits* correspondiente
- El sistema lee la información del gasto y empieza a determinar el gasto máximo de todas las fechas relacionadas al tramo modelado
- El sistema espera a que el usuario seleccione el tramo del caudal principal y guarda su respectivo tramo máximo
- El sistema espera los Caudales tributarios y los suma al gasto máximo

El sistema construye el archivo de Flujo para *HEC-RAS* (Fig. 3.10).

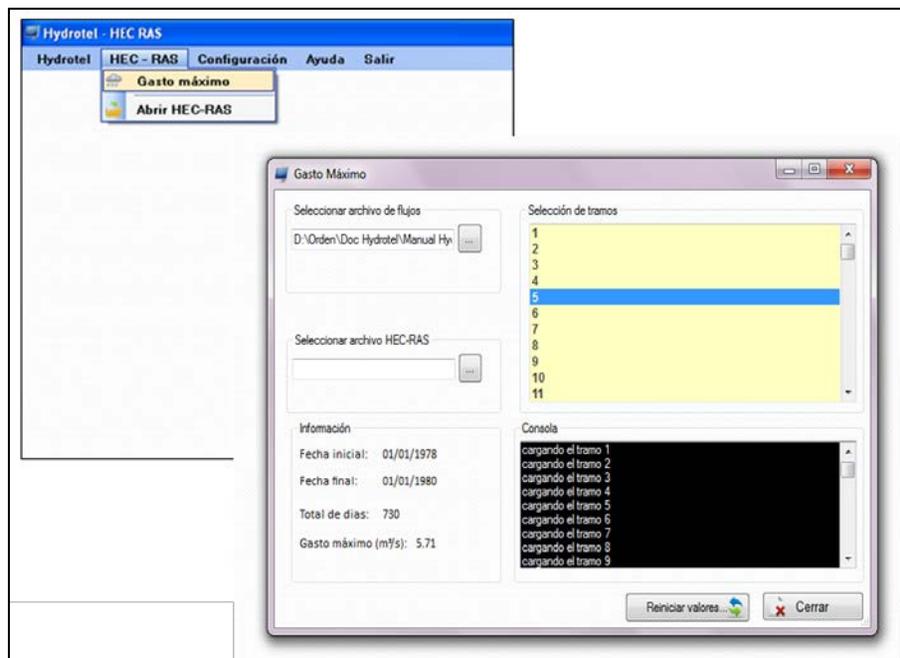


Figura 3.10 Determinación de caudal máximo en el tramo de río seleccionado

3.4 Configuración de resultados

Dentro de la interfaz de usuario es necesario introducir las rutas de acceso físicas (Fig. 3.11) para que el sistema sea capaz de ejecutar el modelo hidrológico e hidráulico, este procedimiento sólo se realiza una vez y la interfaz de usuario guarda estos datos parámetros que estarán disponibles cada vez que el usuario necesite ejecutar algún modelo.

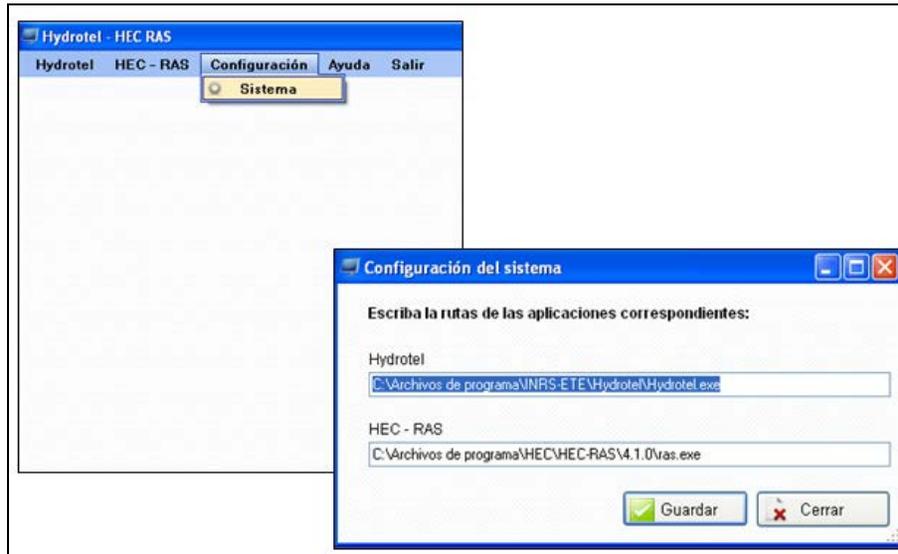


Figura 3.11 Pantalla de configuración del sistema

En caso contrario la aplicación no será capaz de realizar las acciones de “Abrir Hydrotel” y “Abrir HEC-RAS”.

Conclusiones y recomendaciones

Se ha abordado un problema de gran alcance y de problemática compleja, aplicando un instrumento de soporte para la abordar la problemática de inundaciones del río Yautepec, que incorpora en forma coordinada el uso de modelos hidrológicos distribuidos y el de modelos de tránsito de avenidas. La solución adoptada permite el tratamiento de los dos modelos coordinados mediante una interfaz de usuario programada en el lenguaje C#. Los modelos coordinados, cuenta además con alto detalle en los datos de entrada y la información de salida.

En la modelación hidrológica del río Yautepec se tuvieron resultados aceptables en la calibración si se considera que el objetivo de la modelación es reproducir los eventos extremos de escurrimiento, sin embargo el valor de -0.72 del indicador *Nash Sutcliffe*, valor que representa la distribución de la escorrentía en el periodo analizado, es un valor bajo pero representa bien los valores extremos.

La modelación hidráulica se realizó en la plataforma *HEC-RAS* se representaron las secciones topográficas levantadas en el terreno y con el uso del hidrograma resultado de la modelación hidrológica se obtuvo un tirante el cual se compara con el tirante que se obtuvo en las inundaciones históricas.

Una vez concluida la fase de pruebas y mejoras del sistema, se logró conectar los sistemas de *Hydrotel* y *HEC-RAS* mediante una interfaz de usuario, la cual permite la interacción y generación de entradas y salidas para generar pronóstico más precisos de inundación a partir del cálculo del gasto máximo de simulaciones generadas por *Hydrotel*, para generar archivos de flujo que permitan alimentar de información a *HEC-RAS* para el modelado de tránsito de avenidas.

Los resultados de la modelación hidrológica e hidráulica no fueron relevantes, ya que el objetivo del proyecto es comunicar los modelos hidrológico e hidráulico a través de una interfaz de usuario que automatizara el proceso de transferencia de datos del modelo hidrológico al modelo hidráulico.

Con este acoplamiento de modelos es una función desarrollada de gran utilidad y crea la posibilidad de analizar resultados agregados espacialmente, para definir áreas de inundación, con base en los escurrimientos producto de un modelo hidrológico distribuido robusto, y el transido de avenidas modelado en una plataforma con alto grado de confiabilidad.

Se recomienda continuar con la aplicación, ampliando la modelación dentro de la plataforma *HEC-RAS* en cuanto a la definición de áreas de inundación apoyadas en levantamientos topográficos,



esta modelación debe ser mejorada en etapas subsiguientes con la incorporación de funciones nuevas. Asimismo, se debe complementar la interfaz de usuario y mejorar su funcionalidad.

Referencias

1. USGS, 2005. *Shuttle Radar Topography Mission (SRTM3)* de la NASA, Feb 11-22, 2000, <http://edcsns17.cr.usgs.gov/srtmdted>
2. Turcotte, R. et al, 2001. *Determination of the drainage structure of a watershed using a digital elevation model and a digital river and lake network*. Journal of Hydrology 240 (2001) 225–242, www.elsevier.com/locate/jhydrol
3. Fortin, J. P. et al, 2004. *Le Modèle Hydrologique Hydrotel – Bases Théoriques*. INRS ETE, Québec
4. Sanvicente Sánchez, Héctor; Aparicio Mijares, Francisco Javier; 2008. *Análisis y Modelación hidrológica, de calidad del agua y gestión en la región fronteriza México-Estados Unidos*, Instituto Mexicano de Tecnología del Agua.
5. <http://www.hec.usace.army.mil/software/hecras/>
6. <http://www.csharp-station.com/Tutorial.aspx>

Anexos

1. Base Geográfica

Proyecto *ArcMap* (Rasgos vectoriales)

Son seis los proyectos que se integraron para la descripción de la fisiografía y geografía de la cuenca del río Conchos:

- *Bufer 10 Km*
- *Cuerpo de Agua Conchos_cclwgs84*
- *Hidrografía_50k_cclwgs84*
- *Parteaguas Original*
- *Red_1_250K_Conchos*
- *Red1_10000K_Conchos_pl(edición)*

Rasgos matriciales

2. Modelo hidrológico distribuido

- Modelo *Phystel*
- Modelo *Hydrotel*
- Gradientes verticales de temperatura
- Manual de usuario

3. Modelo hidráulico

- Modelo *HEC-Ras*
- Libros *Excel: Datos Yautepec y Resultados Yautepec Simulación*
- *Manual de usuario*

4. Anexo 4. Interfaz de usuario

Manual de usuario

Tabla de contenido

Introducción	108
1. Mapa conceptual de la interfaz de usuario	109
2. Descripción de ventanas del mapa conceptual	110
2.1	Modelo hidrológico <i>Hydrotel</i> 110
2.1.1 Hidrograma	111
2.1.2 Estructurar archivos de salida.....	113
2.1.3 Abrir <i>Hydrotel</i>	115
2.2 Modelo hidráulico <i>HEC-RAS</i>	116
2.2.1 Gasto máximo	117
2.2.2 Abrir <i>HEC-RAS</i>	119
3. Configuración del sistema.....	121

Introducción

El presente manual de usuario tiene la intención de proporcionar los elementos para operar la aplicación de escritorio desarrollada en el lenguaje de programación C# .net Framework 3.5, realizada para comunicar los modelos hidrológicos *Hydrotel* y un sistema que modela el flujo de agua en ríos y canales (*HEC-RAS*).

Para lograr esto es necesario, recopilar, analizar e interpretar los datos de entrada y de salida que generan los modelos *Hydrotel* y *HEC-RAS*, esto con el fin de enlazar estos dos modelos, de tal manera que el usuario final sin muchos conocimientos en el área de hidrología pudiera de forma sencilla, rápida y eficiente operar los modelos hidrológicos e hidráulicos para pronosticar de forma más precisa eventos de inundación en un área en específico.

1. Mapa conceptual de la interfaz de usuario

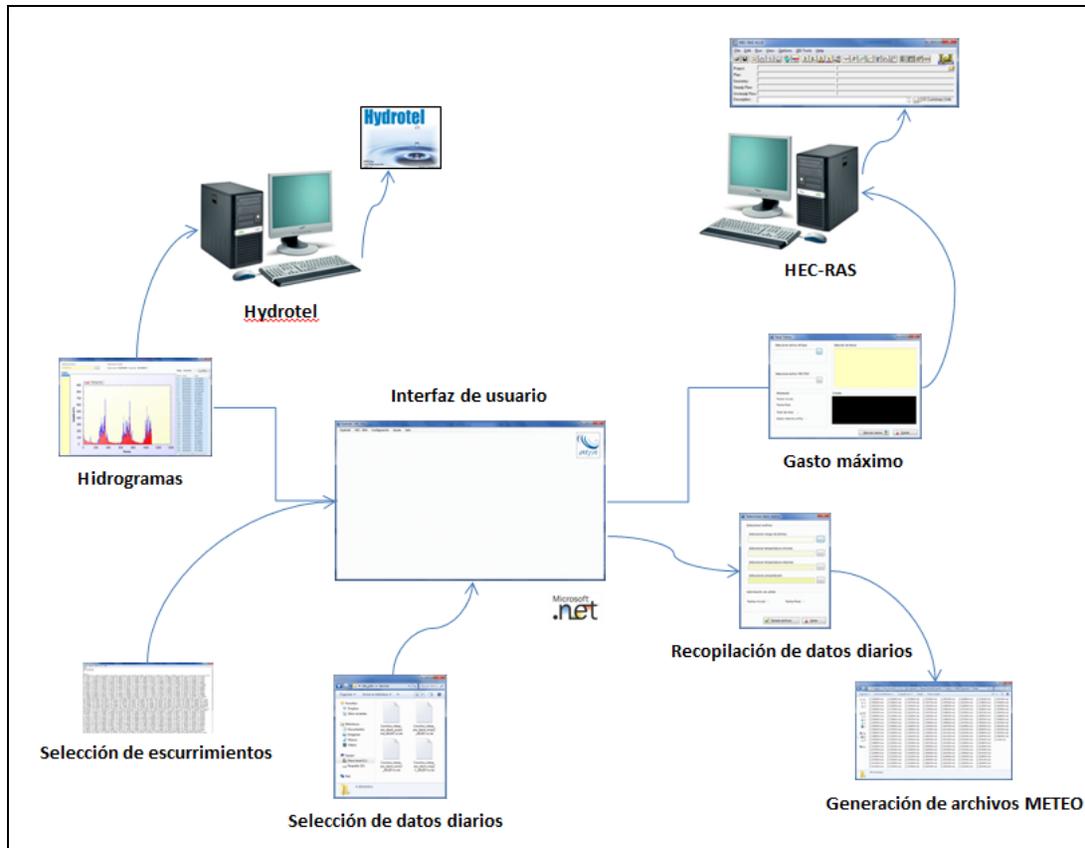


Figura 1.1 Mapa conceptual de la interfaz de usuario

2. Descripción de ventanas del mapa conceptual

La figura 2.1, muestra un mapa con la conceptualización y el funcionamiento de la interfaz de usuario.

Dentro de la interfaz de usuario existe una barra de menús (fig. 2.2) que contiene a su vez, un conjunto de submenús que tienen relación con el modelo hidrológico *Hydrotel* y, el modelo hidráulico *HEC-RAS*, la configuración de las rutas de cada uno de los modelos y una ayuda al usuario final del sistema.



Figura 2.1 Barra de menús de la interfaz de usuario

2.1 Modelo hidrológico *Hydrotel*

La figura 3.1 contiene un conjunto de opciones que ofrecen al usuario el manejo de archivos de entrada y salida que el modelo hidrológico *Hydrotel* necesita y también ofrece la opción de ejecutar *Hydrotel* desde la interfaz de usuario, cuando se ejecuta la opción *Abrir Hydrotel*, automáticamente se desprenderán una serie de pantallas indicando al usuario que es lo que necesita realizar dentro del programa *Hydrotel* para llevar a cabo la simulación.

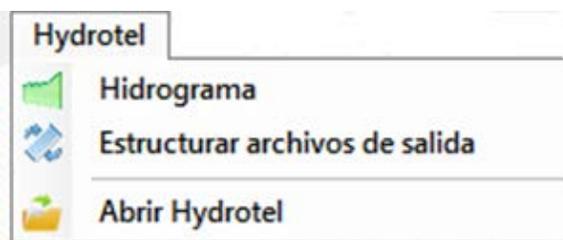


Figura 2.2 Opciones que contiene el submenú *Hydrotel*

2.1.1 Hidrograma

Para generar un hidrograma que represente los diversos gastos a partir de un determinado periodo de tiempo mediante un estudio de un periodo en específico, es necesario alimentar el sistema con un archivo con extensión *.txt que contenga datos acerca de precipitación, dicho archivo debe ser un archivo de salida generado por *Hydrotel*, el cual contiene información de los flujos a lo largo del periodo de tiempo del estudio.

Para mostrar un hidrograma en pantalla es necesario seleccionar que el usuario examine y cargue el archivo, para realizar esta acción el usuario realiza un evento clic sobre el botón  que se encuentra en la sección *Seleccionar archivo* que se encuentra dentro de la pantalla Hidrograma (fig. 2.3). El sistema automáticamente valida el archivo que contenga una estructura de *Hydrotel* para que sea leído correctamente. Si el archivo es correcto, se cargan los tramos encontrados en el archivo dentro de una lista en la sección *tramos* y envía un mensaje al usuario indicando que debe de seleccionar el tramo que desea ver en pantalla, una vez seleccionado el tramo el sistema devuelve un hidrograma con los datos de precipitación contenidos en el archivo. En la sección *Información de salida* se observa el rango de fechas que contiene el archivo que ha sido seleccionado.

Una vez representados los datos de precipitación en el hidrograma, el sistema proporciona un componente para la selección de una fecha, en caso que el usuario desee encontrar un punto en específico basta con seleccionar la fecha y realizar clic con el botón derecho del mouse en el botón Buscar.

Se puede apreciar mejor el flujo de información mediante el siguiente algoritmo:

- El usuario inicia la aplicación, y el sistema espera el archivo de simulación de *Hydrotel* como archivo de entrada.

- El sistema lee los datos de entrada del archivo de simulación generado por *Hydrotel*.
- El sistema determina el rango de *debits* e indica al usuario por medio de un componente *ListBox*, los tramos que puede seleccionar para ser graficados
- El usuario selecciona el tramo de interés.
- El sistema lee e interpreta el tramo seleccionado, y determina la entrada de '*Debits*' correspondiente.
- El sistema lee la información del gasto y empieza a graficar por medio de una librería de C# de gráficos (*ZedGraph*) los gastos de ese tramo en el periodo de tiempo del estudio mediante el siguiente ciclo:

Mientras (Existan días a graficar){

Imprime línea de grafico escalada al valor de gasto correspondiente al día leído

Contador se incrementa en uno para determinar el número de días computados

}

- El sistema termina de graficar y presenta en pantalla los resultados de la gráfica.

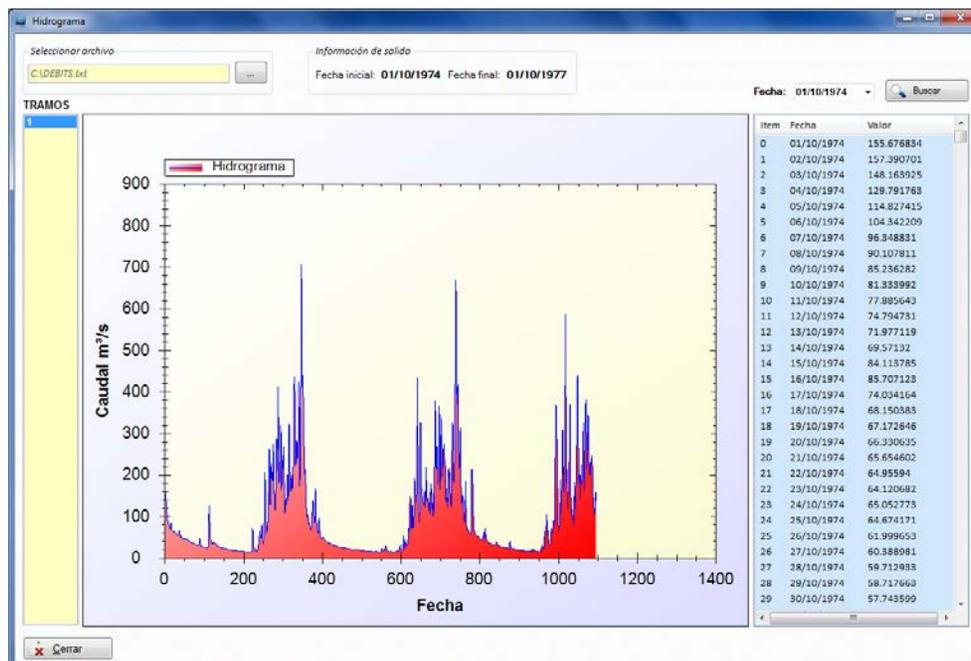


Figura 2.3 Pantalla hidrograma

2.1.2 Estructurar archivos de salida

Para generar los archivos con extensión **.met*, que serán consumidos por la simulación en *Hydrotel*, son necesarios archivos con extensión **.txt* que contenga datos diarios de temperatura mínima, máxima, precipitación y una serie de fechas, estos cuatro archivos son requeridos en la pantalla de *Estructurar archivos de salida* (fig. 2.4) y son seleccionados con su respectivo botón  con el fin de interpretar y estructurar estos datos de tal forma que su sintaxis sea en su totalidad compatible en la manera en que son leídos en el modelo hidrológico *Hydrotel*.

Una vez ingresados estos archivos el usuario realiza el evento (clic en el botón Generar archivos), el sistema valida que los archivos contengan datos numéricos y que sean archivos válidos, si es así, comienza a interpretar y transformar estos datos en una estructura compatible a *Hydrotel* generando dentro de una carpeta llamada *Meteo* archivos con la nueva estructura.

Se puede apreciar mejor el flujo de información mediante el siguiente algoritmo:

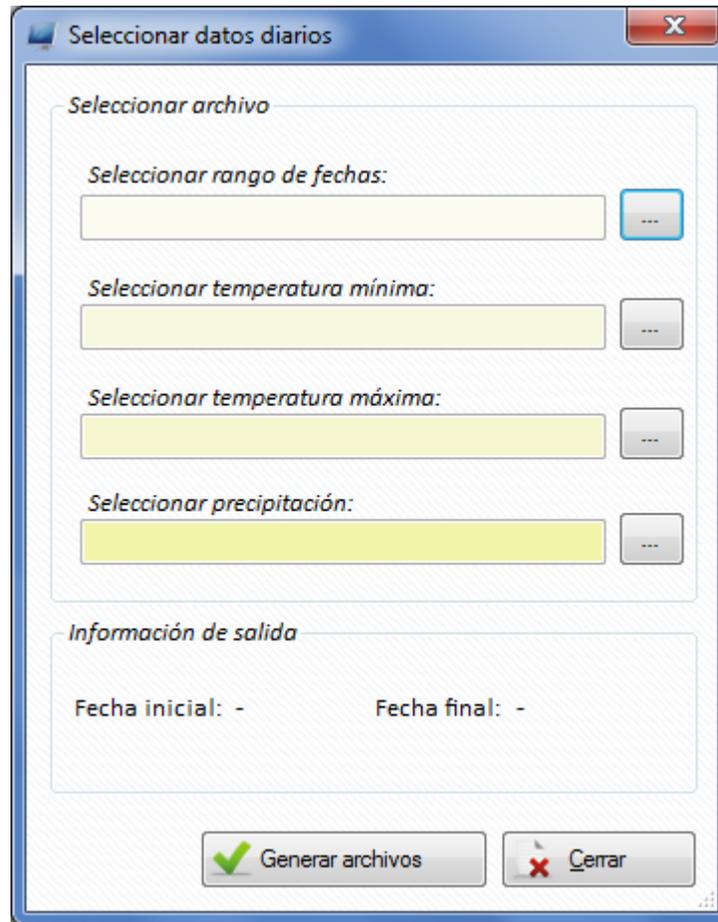
- El usuario inicia la aplicación, y el sistema espera los archivos de fechas, temperatura máxima, mínima así como su respectiva precipitación.
- El sistema lee los datos de entrada de los archivos.
- El sistema construye y relaciona los datos de los archivos para generar una salida compatible para *Hydrotel* mediante el siguiente ciclo:

Mientras (Haya Fechas Disponibles) {

Concatenar la FECHA + TMIN +TMAX + PRECIPITACION

Contador en incrementos de 1+1 }

- El sistema genera un conjunto de archivos con extensión **.met* dentro de una carpeta llamada *Meteo*.



Seleccionar datos diarios

Seleccionar archivo

Seleccionar rango de fechas:

Seleccionar temperatura mínima:

Seleccionar temperatura máxima:

Seleccionar precipitación:

Información de salida

Fecha inicial: - Fecha final: -

Generar archivos Cerrar

Figura 2.4 Pantalla Estructurar archivos de salida

2.1.3 Abrir *Hydrotel*

La acción que debe realiza el usuario es, clic con el botón derecho del mouse sobre la opción (fig. 2.5) *Abrir Hydrotel*, el sistema automáticamente ejecuta dicho programa sobre la aplicación siempre y cuando éste se haya ingresado por lo menos una vez la ruta en el campo *Hydrotel* en la pantalla de *configuración del sistema* (fig. 3.1) y la aplicación *Hydrotel* haya sido instalada en la computadora.

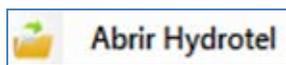


Figura 2.5 Submenú Abrir Hydrotel

En caso de que la ruta no sea correcta o el programa no haya sido instalado en la computadora, el sistema envía un mensaje de error (fig. 2.5).

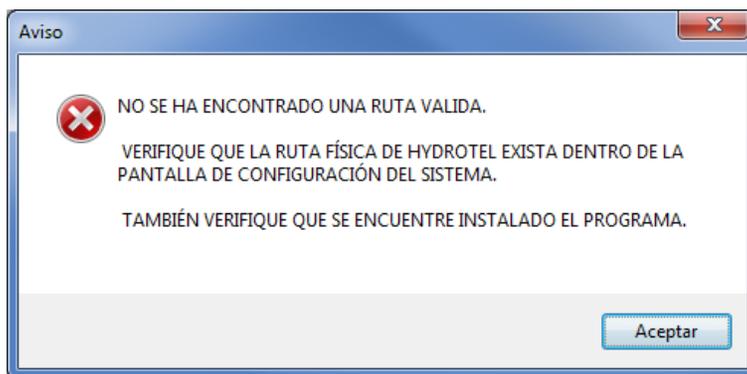


Figura 2.5 Mensaje de error al no encontrar la aplicación Hydrotel

Si la dirección es correcta y el evento clic sobre el botón Abrir *Hydrotel* (fig. 2.5) es realizado, la aplicación ejecuta la aplicación (fig. 2.6).

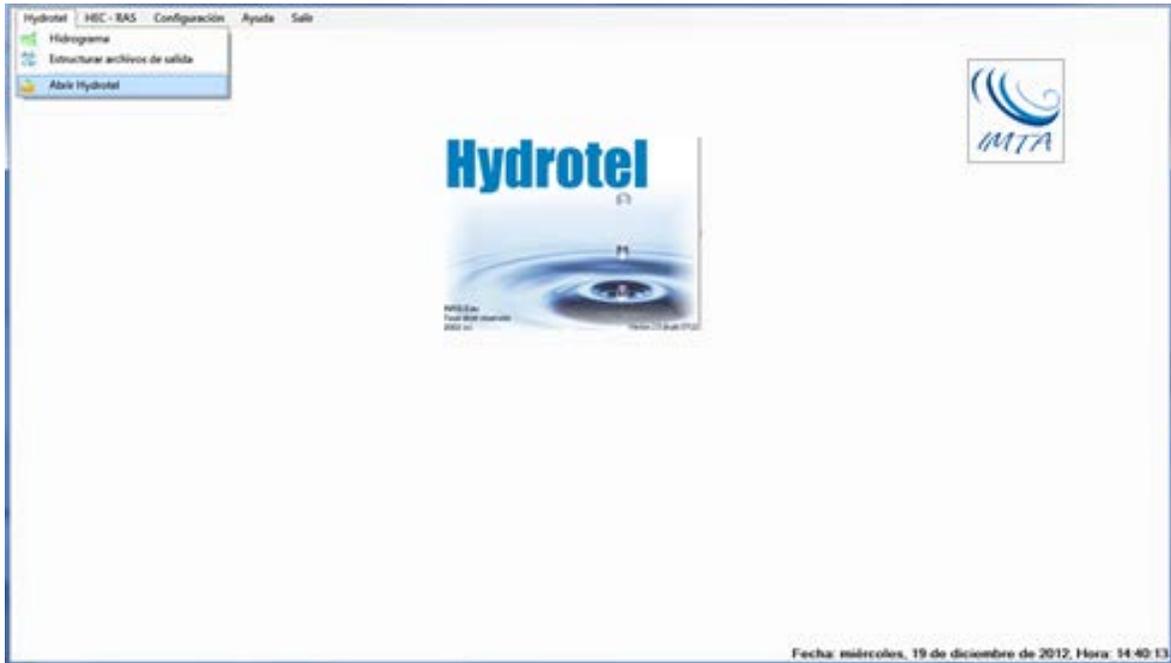


Figura 2.6 Ejecución de Hydrotel

2.2 Modelo hidráulico *HEC-RAS*

A continuación se presentan en la figura 2.7 las opciones que ofrece la interfaz de usuario para el modelo hidráulico *HEC-RAS*.



Figura 2.7 Opciones que contiene el submenú HEC – RAS

2.2.1 Gasto máximo

Para generar el archivo que será consumido por la simulación de *HEC-RAS*, es necesario que el sistema en primer lugar lea un archivo con extensión *.txt, éste archivo debe de contener una serie de **datos que correspondan a precipitación**, una vez ingresado al sistema este archivo, se ejecuta un conjunto de instrucciones, es decir, la interfaz interpreta los datos y crea como salida un archivo nuevo que servirá como entrada para el modelo hidráulico, de tal forma que su sintaxis sea en su totalidad compatible en el lenguaje *HEC-RAS*.

Para empezar a generar el cálculo máximo sobre un archivo de simulación, es necesario que el usuario examine y cargue el archivo, para realizar esta acción el usuario realiza un clic sobre el botón  que se encuentra en la pantalla de Gasto máximo (figura 2.8) en la sección *seleccionar archivo de flujos*. Una vez seleccionado el archivo, el usuario debe dar clic en un elemento de la lista que contiene todos los tramos, el primer gasto seleccionado, se considera como el gasto principal y los tramos seleccionados posteriormente, se interpretan como gastos tributarios, estas acciones se realizan en la sección *Selección de tramos* de la figura mencionada con anterioridad.

Al término del proceso de selección del gasto principal y los gastos tributarios, el usuario tendrá que seleccionar el archivo de flujo de *HEC-RAS* para vaciar la información calculada y seleccionada por el usuario, realizando en este proceso el cálculo de los gastos máximos de cada tramo seleccionado (gasto principal y/o gastos tributarios), esta acción se realiza en la sección *Seleccionar archivo HEC-RAS*, sobre el botón  en la figura 2.8.

Las acciones que realiza el usuario se observan en la bitácora de las actividades del sistema, (sección *consola* de la figura 2.8), la cual contendrá las líneas de cada acción realizada.

Se puede apreciar mejor el flujo de información mediante el siguiente algoritmo:

- El usuario inicia la aplicación, y el sistema espera el archivo de simulación de *Hydrotel* de entrada.
- El sistema lee los datos de entrada del archivo de simulación generado por *Hydrotel*.
- El sistema determina el rango de '*DEBITS*' e indica al usuario por medio de un *ListBox* los tramos que puede seleccionar para ser graficados.
- El usuario selecciona el tramo de interés
- El sistema lee e interpreta el tramo seleccionado, y determina la entrada de *Debits* correspondiente.
- El sistema lee la información del gasto y empieza a determinar el gasto máximo de todas las fechas relacionadas al tramo modelado:

```
Mientras (Existan días a graficar){  
  Si el tramo inicial >= Tramo Actual{  
    Guarda Gasto Máximo  
  }  
  Contador se incrementa en uno para determinar el número de días computados  
}
```

- El sistema espera a que el usuario seleccione el tramo del caudal principal y guarda su respectivo tramo máximo.
- El sistema espera los Caudales tributarios y los suma al gasto máximo
- El sistema construye el archivo de Flujo para *HEC-RAS*.

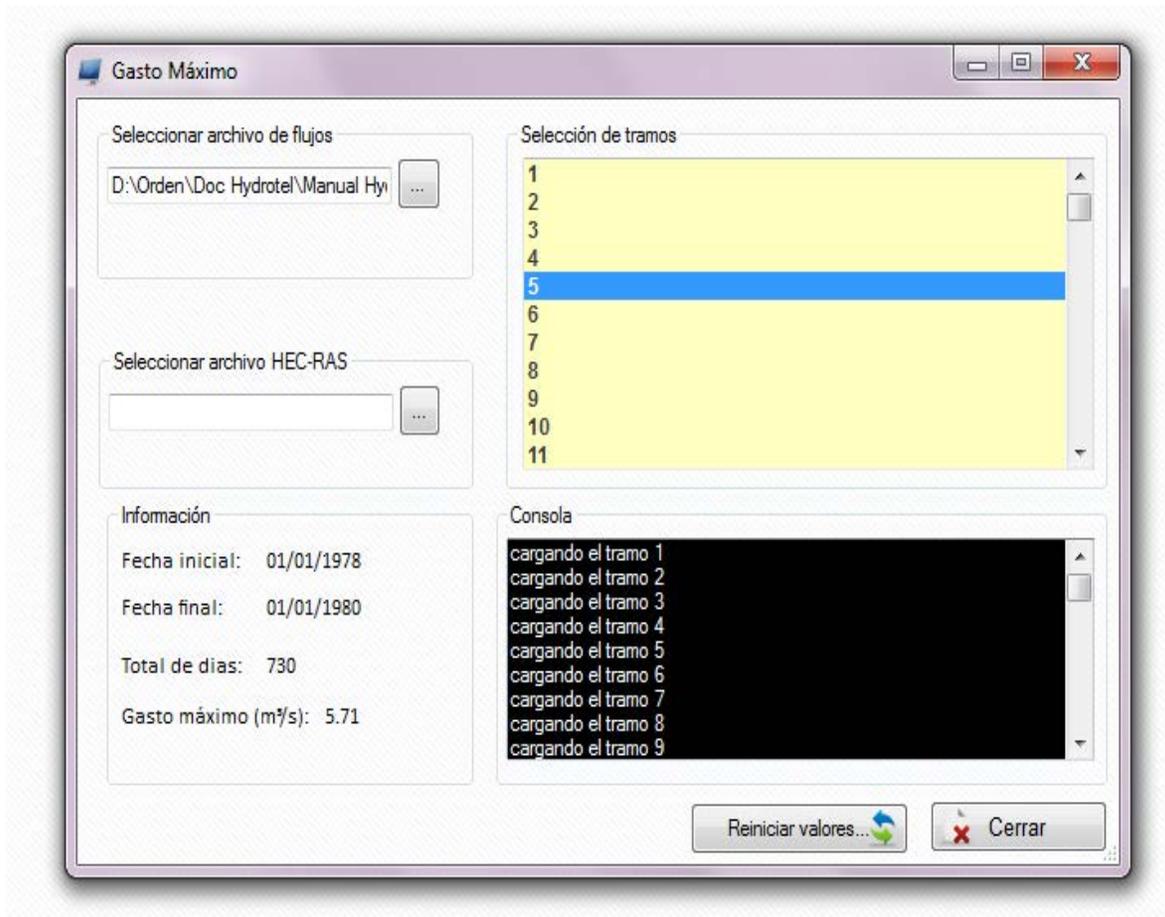


Figura 2.8 Cálculo de Gasto Máximo por Tramo

2.2.2 Abrir HEC-RAS

La acción que debe realizar el usuario es, clic con el botón derecho del mouse sobre la opción *Abrir HEC-RAS* (fig. 2.9), el sistema automáticamente ejecuta dicho programa sobre la aplicación siempre y cuando éste se haya ingresado por lo menos una vez la ruta en el campo *HEC-RAS* en la pantalla de *configuración del sistema* (fig. 3.1) y la aplicación *HEC-RAS* haya sido instalada en la computadora.



Figura 2.9 Submenú Abrir HEC-RAS

Si la dirección es correcta y el evento clic sobre el botón Abrir *HEC-RAS* (fig. 2.9) es realizado, la aplicación ejecuta la aplicación (fig. 2.10).

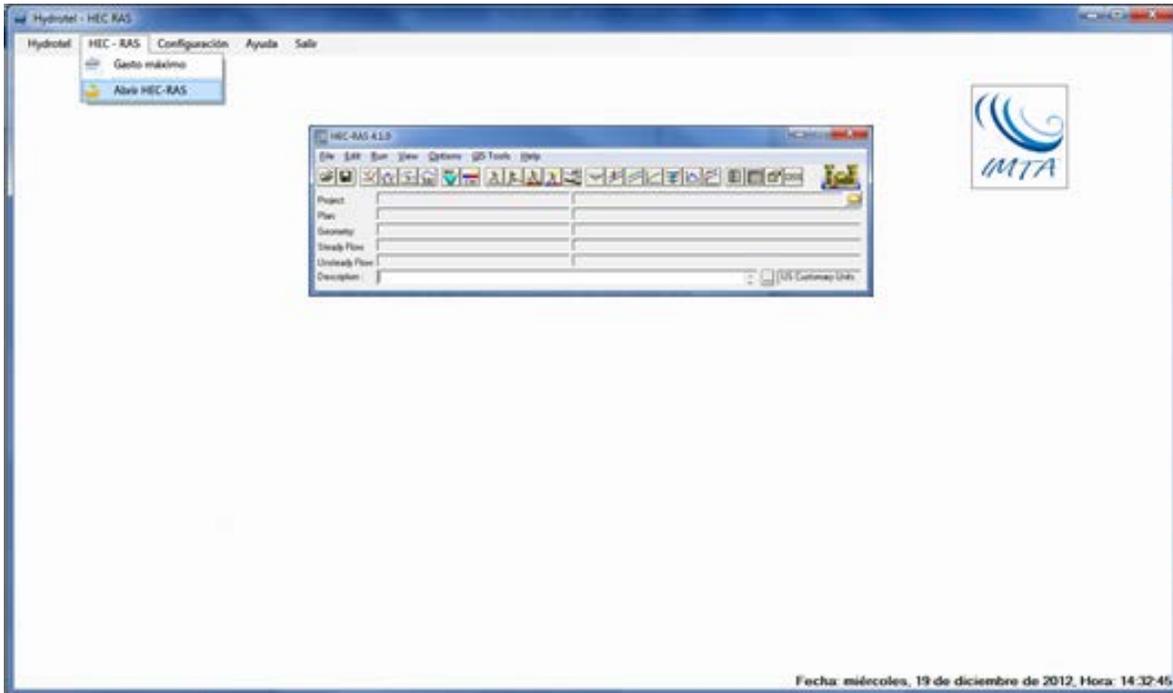


Figura 2.10 Ejecución de HEC-RAS

En caso de que la ruta no sea correcta o el programa no haya sido instalado en la computadora, el sistema envía un mensaje de error (fig. 2.11).

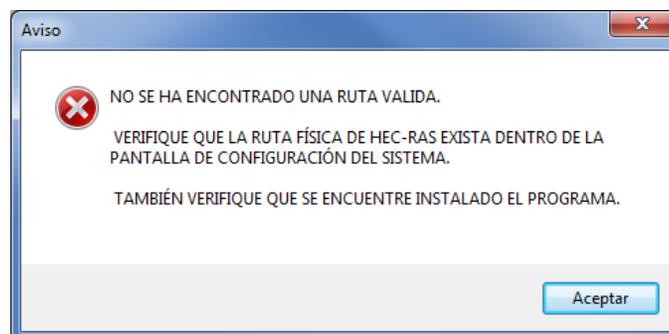


Figura 2.11 Mensaje de error al no encontrar la aplicación HEC-RAS

3. Configuración del sistema

Dentro de la interfaz de usuario es necesario introducir las rutas de acceso físicas (Fig. 3.1) para que el sistema sea capaz de ejecutar el modelo hidrológico e hidráulico, este procedimiento sólo se realiza una vez y la interfaz de usuario guarda estos datos parámetros que estarán disponibles cada vez que el usuario necesite ejecutar algún modelo.

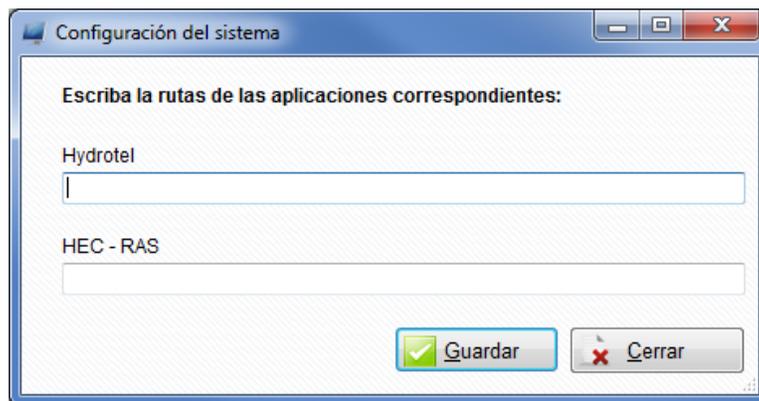


Figura 3.1 Pantalla de configuración del sistema

En caso contrario la aplicación no será capaz de realizar las acciones de *Abrir Hydrotel* (figura 2.5) y *Abrir HEC-RAS* (Fig. 2.9).

Y los resultados?

Código fuente

El código fuente de la interfaz de usuario, se describe un conjunto de líneas de texto el cual conforman las instrucciones que debe seguir la computadora para ejecutar dicho programa. El código fuente está escrito en el lenguaje de programación C# Framework 3.5, cabe mencionar que este código no es directamente ejecutable por la computadora, sino que es traducido a otro lenguaje (lenguaje máquina o código objeto) de tal manera que la computadora sí pueda ejecutar. Para esta traducción se usan los llamados compiladores, ensambladores, intérpretes y otros sistemas de traducción.

Contenido

Principal.cs	124
Configuracion.cs	127
Cálculo de gasto máximo qMax.cs	130
DatosDiarios.cs	138
Hidrograma.cs	147

Principal.cs

La pantalla principal está compuesta por una barra de título que se contiene a la vez cinco submenús *Hydrotel*, *HEC-RAS*, *Configuración*, *Ayuda* y *Salir*, la Figura 1 muestra dicha pantalla, además de contar con los menús anteriormente mencionados también proporciona al usuario una etiqueta que contiene el día y la hora actual.

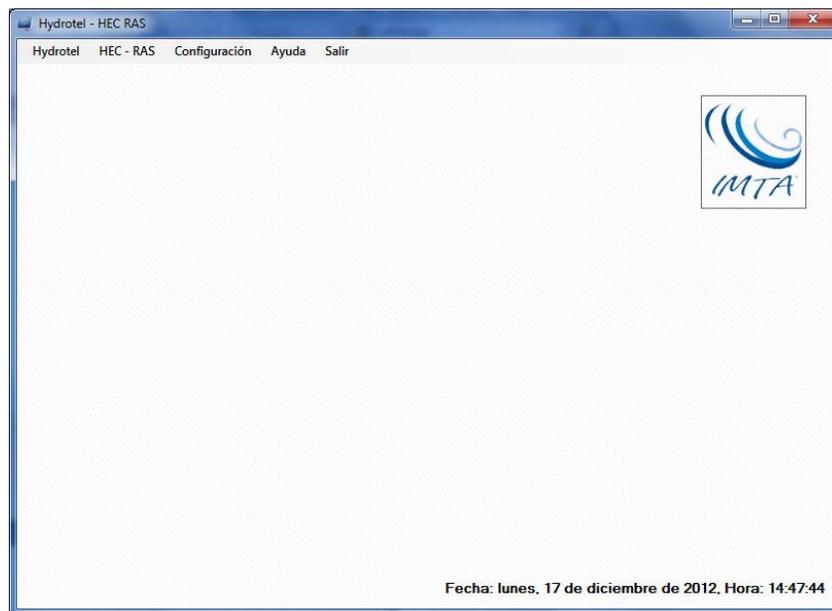


Figura 1. Pantalla principal

Gran parte del código se ejecuta en los eventos “click” que realiza el usuario en sus diferentes submenús. Existe por lo menos una instanciación a un objeto que es creado para el llamado a las diferentes pantallas que provee la interfaz de usuario.

A continuación se describe muestra y el código fuente que realiza toda la funcionalidad dentro de la pantalla principal.cs.

```

public partial class Principal : Form
{
    public Principal()
    {
        //***** Método InitializeComponent() *****
        //Metodo que define en el Designer.cs los controles que ha definido en el diseñador del formulario
        InitializeComponent();
    }

    private void acercaDeToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Utilizando = new para crear objetos e invocar un constructor en éste caso, invocamos
        //al formulario AboutBox1
        AboutBox1 show = new AboutBox1();
        show.ShowDialog();
    }

    private void ayudaToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        //Bloque Try - Catch
        //El bloque try contiene el código protegido que puede causar la excepción.
        //Este bloque se ejecuta hasta que se produce una excepción o hasta completarse satisfactoriamente.

        try
        {
            string path = System.Windows.Forms.Application.StartupPath;
            string Ruta = path + "\\Manual.pdf";

            //Abre el archivo en formato pdf llamado Manual.pdf
            System.Diagnostics.Process.Start(Ruta);
        }
        catch (InvalidCastException exce)
        {
            MessageBox.Show("Verifique que exista el Archivo.\nInstale Adobe Acrobat Reader.\n" + exce, "Aviso",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

    private void datosEntradaToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //Utilizando = new para crear objetos e invocar un constructor en éste caso, invocamos
        //al formulario Qmax() que contiene procedimientos para el calculo de gasto máximo de una sección de tramos.
        Qmax show = new Qmax();
        show.ShowDialog();
    }
}

```

```

private void datosEntradaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Utilizando = new para crear objetos e invocar un constructor en éste caso, invocamos
    //al formulario Qmax() que contiene procedimientos para el calculo de gasto máximo de una sección de tramos.
    Qmax show = new Qmax();
    show.ShowDialog();
}

private void sistemaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Utilizando = new para crear objetos e invocar un constructor en éste caso, invocamos
    //al formulario Configuracion donde se definirá la ruta física de los modelos hidrológicos Hydrotel y HEC-RAS.
    Configuracion show = new Configuracion();
    show.ShowDialog();
}

private void hidrogramaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Utilizando = new para crear objetos e invocar un constructor en éste caso, invocamos
    //al formulario SacarTramos el cual contiene rutinas para la elaboración de hidrogramas.
    SacarTramos show = new SacarTramos();
    show.ShowDialog();
}

private void estructurarArchivosDesalidaToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Utilizando = new para crear objetos e invocar un constructor en éste caso, invocamos
    //al formulario DatosEntrada el cual contiene rutinas para transformar los datos de temperatura mínima, máxima y precipitación,
    //a una estructura que hydrotel necesita para elaborar simulaciones.
    DatosEntrada show = new DatosEntrada();
    show.ShowDialog();
}

private void abrirHydrotelToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Bloque Try - Catch
    //El bloque try contiene el código protegido que puede causar la excepción.
    //Este bloque se ejecuta hasta que se produce una excepción o hasta completarse satisfactoriamente.
    try
    {
        string DireccionFile = urls("Hydrotel");
        //Ejecuta el método urls() con una variable hydrotel para obtener la dirección física con el nombre de Hydrotel
        System.Diagnostics.Process.Start(DireccionFile);
    }
    catch (Exception p)
    {
        //En caso de no encontrar la ruta de hydrotel, enviar un mensaje al usuario mostrando en pantalla el error.
        MessageBox.Show("" + p, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private string urls(string viene)
{
    int counter = 0;
    string line;
    string cadena = "", regresa = "";
    bool EsNumero = false;
    float Numero = 0;

    //Obtiene la ruta de acceso del archivo ejecutable que inició la aplicación, sin incluir el nombre del archivo ejecutable.
    string path = System.Windows.Forms.Application.StartupPath;

    string Ruta = "";

    //Asigna por medio de la variable "viene" el nombre del archivo en formato txt para ejecutar el archivo ejecutable que el
    //Archivo txt contenga.
    if (viene == "Hydrotel") Ruta = path + "\\directoryhydrotel.txt";
    if (viene == "HECRAS") Ruta = path + "\\directoryhecras.txt";

    //Lee una línea de caracteres de la secuencia actual y devuelve los datos como una cadena.
    System.IO.StreamReader file = new System.IO.StreamReader(Ruta);
    while ((line = file.ReadLine()) != null)
    {
        cadena = line;
        if (counter == 0)
        {
            //Valida que la cadena contenga caracteres alfanuméricos si el caso es correcto... devuelve la cadena de texto que contiene el archivo txt

            EsNumero = float.TryParse(cadena, out Numero);
            if (EsNumero == false) regresa = cadena;
        }
    }
    file.Close(); //Cierra el documento en formato *.txt
    return regresa; //regresa la cadena con la ruta física.
}

```

```

private void abrirHECRASToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Bloque Try - Catch
    //El bloque try contiene el código protegido que puede causar la excepción.
    //Este bloque se ejecuta hasta que se produce una excepción o hasta completarse satisfactoriamente.
    try
    {
        string DireccionFile = urls("HECRAS");
        //Ejecuta el método urls() con una variable HECRAS para obtener la dirección física con el nombre de Hydrotel
        System.Diagnostics.Process.Start(DireccionFile);
    }
    catch (Exception p)
    {
        //En caso de no encontrar la ruta de hydrotel, enviar un mensaje al usuario mostrando en pantalla el error.
        MessageBox.Show(" " + p, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void Principal_Load(object sender, EventArgs e)
{
    //Obtiene la fecha y la hora para mostrar en la pantalla de bienvenida.
    label1.Text = "Fecha: " + DateTime.Now.ToLongDateString() + ", Hora: " + DateTime.Now.ToString("HH:mm:ss");
}

private void salirToolStripMenuItem_Click(object sender, EventArgs e)
{
    //Muestra un mensaje preguntando al usuario si desea salir del sistema.
    DialogResult reply = MessageBox.Show("¿Desea salir del sistema?", "Salir del sistema...", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (reply == DialogResult.No)
    {
        //Si el usuario oprime en el botón No, la condición no realiza ninguna acción.
        return;
    }
    else
    {
        //En caso contrario, cierra completamente la aplicación
        this.Close();
        Application.Exit();
    }
}
}
}
}

```

Configuracion.cs

La pantalla Configuración del sistema se compone de dos cajas de texto (Hydrotel, HEC-RAS) y dos botones (Guardad y Cerrar), Figura 2, las instrucciones que contiene el botón guardar son para la creación y guardado de archivos en formato *.txt, las cadenas a guardar son las rutas donde se encuentran instalados los programas Hydrotel y HECRAS.

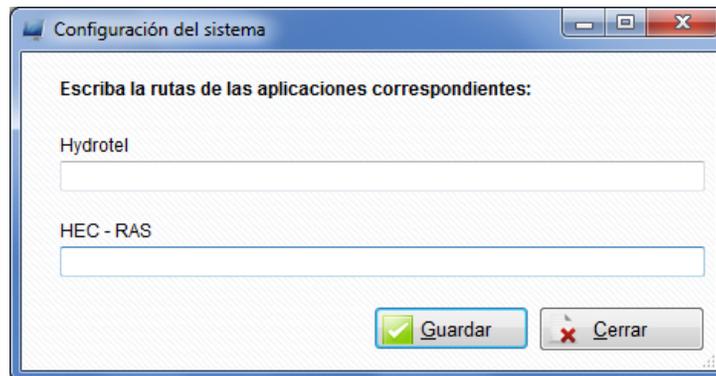


Figura 2. Pantalla de configuración

A continuación se describe el código fuente para la creación de archivos de textos planos para guardar rutas físicas de los modelos hidrológicos (Hydrotel, HEC-RAS) con el fin de ejecutar estos programas automáticamente desde la interfaz de usuario.

```

public partial class Configuracion : Form
{
    public Configuracion()
    {
        //***** Método InitializeComponent() *****
        //Metodo que define en el Designer.cs los controles que ha definido en el diseñador del formulario
        InitializeComponent();
    }

    private void button100_Click(object sender, EventArgs e)
    {
        //Botón cerrar
        //Oculta el formulario, es equivalente a establecer la propiedad Visible en false.
        this.Hide();

        //Libera recursos, en éste caso libera los recursos del formulario Configuracion.cs.
        this.Dispose();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        //Ejecuta una condición, en caso de que las cajas de texto esten vacias, se asigna en cada caja de texto
        //una cadena de texto 'NOURL' indicando al usuario que la caja de texto no contiene una dirección.
        if (textBox7.Text == "") textBox7.Text = "NOURL";
        if (textBox9.Text == "") textBox9.Text = "NOURL";

        //Ejecuta el método escribeEnArchivos() enviando una variable Hydrotel y HECRAS, para que el método
        //Escriba en los archivos correspondientes la dirección física de cada programa
        escribeEnArchivos("Hydrotel");
        escribeEnArchivos("HECRAS");

        //Se envia un mensaje al usuario indicando que se ha registrado exitosamente la ruta física en los archivos de texto
        MessageBox.Show("Se ha registrado exitosamente su información", "Aviso", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        this.Hide();
    }

    //Método que recibe como parametro una variable de tipo string el cual indicará al sistema, hacia donde
    //dirigirse para escribir el archivo de texto.
    private void escribeEnArchivos(string viene)
    {
        //Implementa StreamWriter para escribir los caracteres de una secuencia en una codificación determinada.
        StreamWriter sw = null;
        //Obtiene la ruta del archivo ejecutable que inició la aplicación, sin incluir el nombre del ejecutable.
        string path = System.Windows.Forms.Application.StartupPath;
        string Ruta = "";

        if (viene == "HECRAS") Ruta = path + "\\directoryhecras.txt";
        if (viene == "Hydrotel") Ruta = path + "\\directoryhidrotel.txt";

        //Determina si el archivo existe en la ruta física
        if (File.Exists(Ruta))
        {
            //Crea un archivo *.txt on el nombre directoryhecras.txt o directoryhidrotel.txt
            sw = File.CreateText(Ruta);
            //Escribe en el archivo dependiendo la variable que ha entrado en el método
            if (viene == "HECRAS") sw.WriteLine(textBox9.Text.Trim());
            if (viene == "Hydrotel") sw.WriteLine(textBox7.Text.Trim());
            //Cierra el archivo
            sw.Close();
        }
    }

    //Método load que recibe un parametro para mostrar en pantalla al momento que es invocado el formulario
    //las direcciones físicas tanto de Hydrotel como de HECRAS.

```

```

private void load(string viene)
{
    //Declaración de variables necesarias para interactuar con los archivos de texto
    int counter = 0;
    string line;
    string cadena = "";
    bool EsNumero = false;
    float Numero = 0;

    //Obtiene la ruta del archivo ejecutable que inició la aplicación, sin incluir el nombre del ejecutable.
    string path = System.Windows.Forms.Application.StartupPath;
    string Ruta = "";

    if (viene == "Hydrotel") Ruta = path + "\\directoryhydrotel.txt";
    if (viene == "HECRAS") Ruta = path + "\\directoryhecras.txt";

    //Implementa un TextReader que lee los caracteres de una secuencia de bytes en una codificación determinada.
    System.IO.StreamReader file = new System.IO.StreamReader(Ruta);

    //Comienza a leer el archivo que contiene la ruta física
    while ((line = file.ReadLine()) != null)
    {
        cadena = line;
        if (counter == 0)
        {
            //Valida el archivo contenga caracteres, es decir una dirección física y que no contenga números
            EsNumero = float.TryParse(cadena, out Numero);
            if (EsNumero == false)
            {
                //Una vez leído el archivo de texto asignar a las cajas de texto del sistema la ruta que le corresponde a
                // en cada caja de texto.
                if (viene == "Hydrotel") textBox7.Text = cadena;
                if (viene == "HECRAS") textBox9.Text = cadena;
            }
            else
            {
                //En caso de no encontrar una cadena de caracteres el sistema envía un mensaje de error.
                MessageBox.Show("Error la dirección no es válida.", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
    file.Close();
}

private void Configuracion_Load(object sender, EventArgs e)
{
    //Ejecuta el método load con diferentes variables de entrada.
    load("Hydrotel");
    load("HECRAS");
}
}

```

Cálculo de gasto máximo qMax.cs

A continuación se describe y muestra el código fuente de la interfaz de usuario para el cálculo del gasto máximo, este conjunto de líneas conforman las instrucciones que debe seguir la computadora para ejecutar dicho programa.

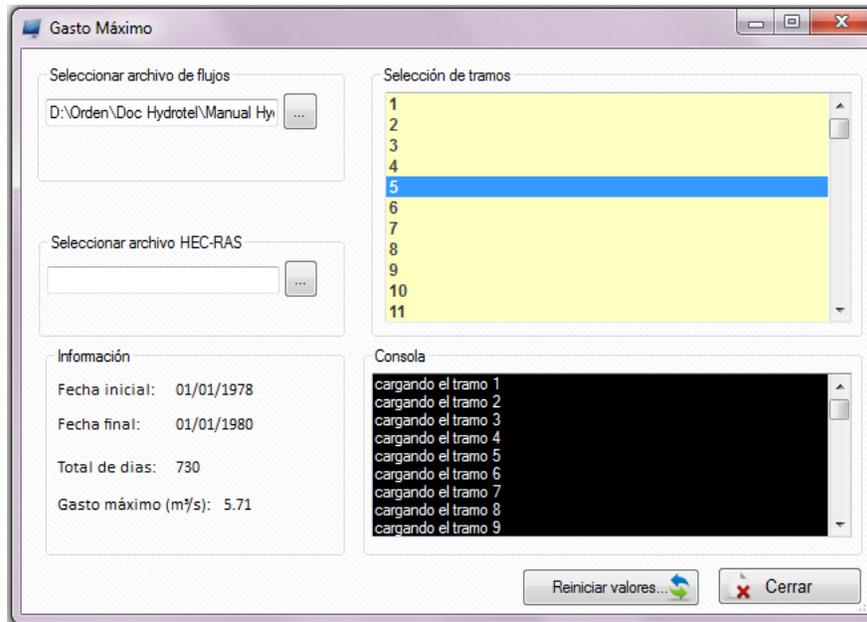


Figura 3. Gasto máximo

```
public partial class Qmax : Form
{
    //Se definen variables que son utilizadas durante la ejecución del programa, las variables están agrupadas por su tipo de dato
    DateTime f1;
    DateTime f2;

    public List<double> ListaTramos = new List<double>();
    public List<double> ListaInformacion = new List<double>();

    public double[] valores;
    public double gastoMaximoTramo = 0;
    public double gastoMaximo = 0;

    public string[,] MisFechas;

    public string FechaMaximoTramo = "";
    public string gastoTributario = "";
    public string FechaMaximo = "";
    public string DireccionText = "";
    public int PrincipalesTributario = 0;
}
```

```

public Qmax()
{
    InitializeComponent();
}

//***** Método button1_Click() *****
//Método del evento click del boton que recupera el archivo de gastos seleccionado por el usuario
private void button1_Click(object sender, EventArgs e)
{
    //Método para limpiar los valores almacenados en las variables
    limpia();

    //Almacena la dirección del componente OpenFileDialog
    DireccionText = TxtSimulacion.Open(openFileDialog1);

    //Si el componente OpenFileDialog contiene caracteres entra a la condición lógica y
    //limpia los componentes listBox y ejecuta el método calculaTramos
    if (DireccionText != "false")
    {
        listBox1.Items.Clear();
        listBox2.Items.Clear();
        textBox1.Text = DireccionText;
        calculaTramos();
    }
}

private void limpia()
{
    // Módulo para limpiar quitar si provoca errores
    gastoMaximoTramo = 0;
    FechaMaximoTramo = "";
    FechaMaximo = "";
    PrincipalOTributario = 0;
    gastoMaximo = 0;
    gastoTributario = "";
    ListaInformacion.Clear();
    ListaTramos.Clear();
    listBox1.Items.Clear();
    listBox2.Items.Clear();
    label3.Text = "-";
    label4.Text = "-";
    label6.Text = "-";
    lMaximoQ.Text = "-";
    textBox2.Text = "";
    textBox1.Text = "";
    // fin modulo para limpiar quitar si provoca errores

    listBox2.Items.Clear();
}

//***** Método calculaTramos() *****
//Método que calcula el numero de tramos existentes en una simulación de hydrotel que pinta los tramos en la interfaz
// para su selección
public void calculaTramos()
{
    int counter = 0;
    string line;
    string[] direccion;
    string cadena = "";
    int numeroTramos = 0;
    bool EsNumero = false;
    float Numero = 0;

    string[] datos;
    string directorio = openFileDialog1.SafeFileName;

    direccion = directorio.Split('.');

    string nombre = direccion[0];
    bool EntroDebit = false;

    //Lee el archivo y lo recorre línea por línea.
    System.IO.StreamReader file = new System.IO.StreamReader(DireccionText);
    while ((line = file.ReadLine()) != null)
    {
        cadena = line;
    }
}

```

```

if (counter == 0) {
    //Valida la primer línea del archivo de texto
    string [] datos1 = cadena.Split(' ');
    EsNumero = float.TryParse(datos1[0].ToString(), out Numero);
    if (EsNumero != true)
    {
        MessageBox.Show("Archivo incorrecto, por favor verifique.", "Error de selección de archivo",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
}
//Coloca el puntero del mouse en modo de espera...
Cursor = Cursors.WaitCursor;

if (counter == 1) { f1 = DateTime.Parse(extraeFecha(counter, cadena)); label3.Text = f1.ToShortDateString();
    label3.Visible = true; }
if (counter == 2)
{
    f2 = DateTime.Parse(extraeFecha(counter, cadena)); label4.Text = f2.ToShortDateString(); label4.Visible = true;
    label6.Text = totalDias(f1, f2).ToString();
}

if (counter == 3)
{
    numeroTramos = int.Parse(line);
    if (numeroTramos != 1)
        numeroTramos = numeroTramos - 1;
}

if (counter > 3 && counter <= numeroTramos + 3)
{
    listBox1.Items.Add("Cargando el tramo " + cadena);
    ListaTramos.Add(double.Parse(cadena));
}

if (counter > (numeroTramos + 3))
{
    datos = cadena.Split(' ');
    EsNumero = float.TryParse(datos[0].ToString(), out Numero);

    if (datos[0].ToString() == "DEBITS")
    {
        //Bandera para indicar que ha encontrado la palabra 'DEBITS'
        EntroDebit = true;
    }
    else
    {
        if (EsNumero == false) EntroDebit = false;
    }
}

if (EsNumero == true)

```

```

        {
            if (EntroDebit == true)
            {
                //Realiza el recorrido en el arreglo y va llenando el componente ListaInformación de los
                //valores encontrados en la sección de DEBITS
                for (int i = 0; i < datos.Length-1; i++)
                {
                    ListaInformacion.Add(double.Parse(datos[i]));
                }
            }
        }

        counter++;
    }
    //cierra el archivo de texto
    file.Close();
    listBox1.Items.Add("Total de tramos: " + numeroTramos.ToString());
    listBox1.Items.Add("Cargada la informacion DEBITS");

    int xx=0;
    valores = new double[ListaInformacion.Count];
    foreach (double element in ListaInformacion)
    {
        valores[xx] = element;
        xx++;
    }

    //Indica el total de los valores encontrados en la lista ListaInformación
    listBox1.Items.Add("Total de Valores:" + valores.Length.ToString());

    foreach (double element in ListaTramos)
    {
        double y = element; // Y ES MI TRAMO
        listBox2.Items.Add(y.ToString());
    }

    //Limpia listas una vez arrojados sus valores dentro de componente listBox2
    ListaInformacion.Clear();
    ListaTramos.Clear();

    //Coloca el puntero del mouse en modo normal
    Cursor = Cursors.Default;
}

//***** Método SaberValores() *****
//Metodo que calcula el rango de valores a partir de un tramo seleccionado

public void SaberValores(int indice, double valortramo)
{
    int Tdias = int.Parse(label16.Text);

```

```

int contador = indice;
int PInicial = 0;
int PFinal = 0;
int resultado = 0;

PFinal = (contador + 1) * Tdias;
if (contador >= 1)
    PInicial = contador * Tdias;
else
    PInicial = 0;

resultado = PFinal - PInicial;
listBox1.Items.Add("TRAMO NUMERO:" + valortramo.ToString());
listBox1.Items.Add("Inicia " + PInicial + " Final " + PFinal + " Total Dias " + resultado.ToString());

MandarFechas(PInicial, PFinal);
}

//***** Método MandarFechas() *****
//Metodo que calcula gasto maximo a partir de un tramo seleccionado en base a sus fechas de captura

private void MandarFechas(int pinicia, int pfinal)
{
    int x = 0;
    double maximoQ = 0;
    int indiceFecha=0;

    MisFechas = new string[3, valores.Length];
    DateTime FechaInicia = DateTime.Parse(label3.Text);

    DateTime FechaRecorre = FechaInicia;
    double ValorFinal = 0;
    for (int i = pinicia; i < pfinal; i++)
    {
        string Fecha = FechaRecorre.ToString("dd/MM/yyyy");
        ValorFinal = valores[i];

        MisFechas[0, x] = x.ToString();
        MisFechas[1, x] = Fecha;
        MisFechas[2, x] = ValorFinal.ToString();

        if (ValorFinal > maximoQ){
            maximoQ = ValorFinal;
            indiceFecha = x;
        }

        FechaRecorre=FechaRecorre.AddDays(1);
        x++;
    }

    gastoMaximoTramo = Math.Round(maximoQ, 2);
}

```

```

FechaMaximoTramo = MisFechas[1, indiceFecha].ToString();
lMaximoQ.Text = Math.Round(maximoQ, 2).ToString();
}

//***** Método extraeFecha() *****
//Método que despliega las fechas de los gastos
private string extraeFecha(int numLinea, string cadena)
{
    string anno = "", dia = "", mes = "";

    string[] fecha = cadena.Split('-');
    anno = fecha[0];
    mes = fecha[1];
    dia = fecha[2];

    string[] separaDosPunto = dia.Split(':');

    string retornaFecha = separaDosPunto[0] + "/" + mes + "/" + anno;

    return retornaFecha;
}

private double totalDias(DateTime f1, DateTime f2)
{
    TimeSpan total = f2 - f1;
    double TotalDays = total.TotalDays;

    return TotalDays;
}

//***** Método listBox2_SelectedIndexChanged() *****
//Identifica el tramo seleccionado por el usuario y lo prepara para calcular su gasto maximo
private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    int Indice = listBox2.SelectedIndex;
    double valor = double.Parse(listBox2.SelectedItem.ToString());
    SaberValores(Indice, valor);

    if (PrincipalOTributario == 0)
    {
        FechaMaximo = FechaMaximoTramo;
        gastoMaximo = gastoMaximoTramo;
        MessageBox.Show("Se definio el Gasto del Caudal Principal: " + gastoMaximo + " con la fecha de: "+FechaMaximo);
        listBox1.Items.Add("Se definio el Gasto del Caudal Principal: " + gastoMaximo);
        PrincipalOTributario++;
    }
    else
    {
        gastoTributario += (gastoMaximoTramo + gastoMaximo) + "";
        MessageBox.Show("Se definio un nuevo gasto tributario: "+gastoTributario);
        listBox1.Items.Add("Se definio un nuevo gasto tributario: " + gastoTributario);
    }
}

```

```

    }
}

//***** Método button3_Click() *****
//Método crea el archivo de Flujo de Hec_Ras en su lenguaje nativo
private void button3_Click(object sender, EventArgs e)
{
    if (gastoMaximo!=0)
    {
        string line;
        string abreHecras;
        StreamWriter destino = null;
        StreamReader fuente = null;
        string guardaHecras;
        double Num;
        string nuevoGastoInsertar;
        int numTributarios =0;
        int contadorTributarios = 0;
        int pusoPrincipal = 0;

        string[] gastosTributarios;

        gastosTributarios = gastoTributario.Split(';');
        numTributarios = gastosTributarios.Length-1;

        abreHecras = TxtSimulacion.Open(openFileDialog1).Trim();

        if (abreHecras != "false")
        {
            textBox2.Text = abreHecras;
            guardaHecras = abreHecras;
            System.IO.File.Copy(abreHecras, abreHecras + ".back", true);
            fuente = new System.IO.StreamReader(abreHecras + ".back");
            destino = File.CreateText(guardaHecras);

            while ((line = fuente.ReadLine()) != null)
            {
                // busca linea para cambiar titulo
                if (line.Substring(0, 8) == "Flow Tit")
                {
                    destino.WriteLine("Flow Title=Gasto del - " + FechaMaximo);
                }
                else if (line.Substring(0, 7) == "Profile")
                {
                    destino.WriteLine("Profile Names=Gasto del - " + FechaMaximo);
                }
                else if (line.Substring(0, 8) == "Number o")
                {
                    destino.WriteLine("Number of Profiles= 1");
                }
                // busca linea para cambiar gasto
                else if (line.Substring(0, 1) == " " || double.TryParse(line.Substring(0, 1), out Num))
                {

```

```

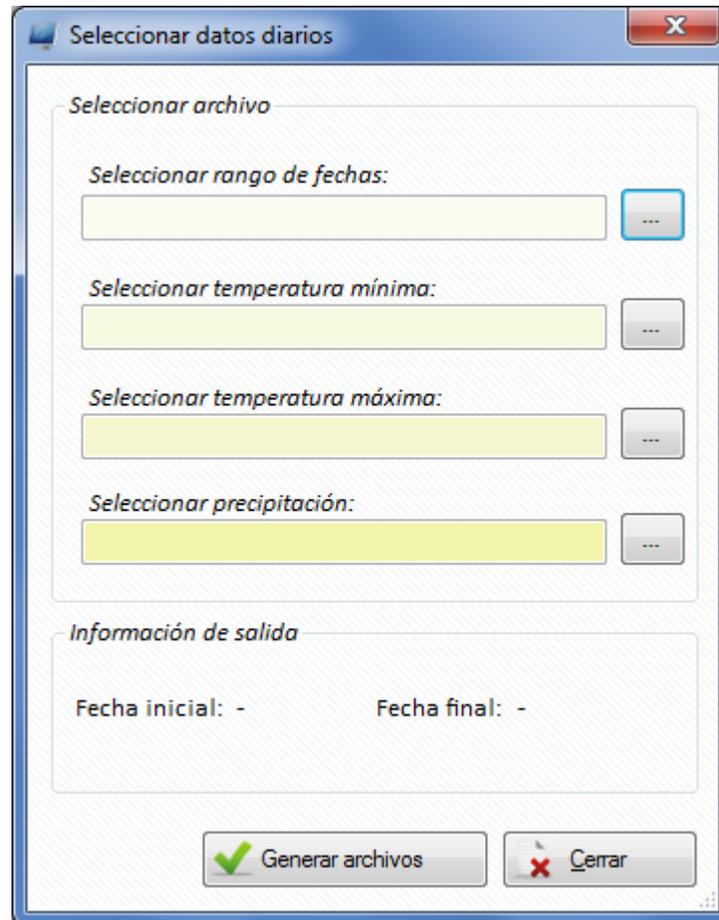
        if (pusoPrincipal == 0)
        {
            //Ponemos los rellenos en blanco para completar bloques de 8 caracteres
            nuevoGastoInsertar = "";
            for (int i = 0; i < 8 - gastoMaximo.ToString().Length; i++)
            {
                nuevoGastoInsertar += " ";
            }
            destino.WriteLine(nuevoGastoInsertar+gastoMaximo.ToString());
            pusoprincipal++;
        }
        else if (contadorTributarios < numTributarios)
        {
            nuevoGastoInsertar = "";
            for (int i = 0; i < 8 - gastosTributarios[contadorTributarios].Length; i++)
            {
                nuevoGastoInsertar += " ";
            }
            destino.WriteLine(nuevoGastoInsertar + gastosTributarios[contadorTributarios].ToString());
            contadorTributarios++;
        }
    }
    else
    {
        MessageBox.Show("Debe seleccionar un archivo de destino.");
    }
}
else
{
    MessageBox.Show("Debe seleccionar un tramo con información disponible");
}
}

private void btnReiniciar_Click(object sender, EventArgs e)
{
}
//***** Método button2_Click() *****
//Método que reinicia el calculo del gasto máximo
private void button2_Click(object sender, EventArgs e)
{
    gastoMaximoTramo = 0;
    FechaMaximoTramo = "";
    FechaMaximo = "";
    PrincipalOTributario = 0;
    gastoMaximo = 0;
    gastoTributario = "";
    listBox1.Items.Clear();
}
private void button100_Click(object sender, EventArgs e)
{
    this.Hide();
    this.Dispose();
}

```

DatosDiarios.cs

La pantalla DatosDiarios.cs contiene un conjunto de instrucciones que recopilan datos acerca de la temperatura máxima, mínima, precipitación y un archivo que contiene un determinado número de fechas.



Selección de datos diarios

Selección de archivo

Selección de rango de fechas:

Selección de temperatura mínima:

Selección de temperatura máxima:

Selección de precipitación:

Información de salida

Fecha inicial: - Fecha final: -

Generar archivos Cerrar

Figura 4. Selección de archivos de precipitación, temperatura máxima, mínima y fechas

A continuación se presenta el código contenido principalmente en el botón Generar archivos. El botón mencionado anteriormente manda a llamar a diferentes métodos para la lectura, transformación y escritura de archivos contenidos en una carpeta llamada Meteo.

```

public partial class DatosEntrada : Form
{
    //Declaración de variables por tipo de dato

    //Lista para recolectar los valores de Fechas, temperatura máxima, mínima y precipitación
    public List<string> ListaInformacion = new List<string>();
    public List<string> ListaValores = new List<string>();
    public List<string> ListaValoresMaxima = new List<string>();
    public List<string> ListaValoresPrecipitacion = new List<string>();

    public string DireccionText = "";
    public string[] ListaFechas;
    public string[] ListaTempMinima; // datos de temperatura mínima
    public string[] ListaTempMaxima; // datos de temperatura máxima
    public string[] ListaTempPrecipitacion; // datos de precipitación

    public DatosEntrada()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        //Calcula fechas si el componente OpenFileDialog arrojo una dirección física
        DireccionText = TxtSimulacion.Open(openFileDialog1);

        if (DireccionText != "false")
        {
            //Método que calcula fechas de los datos obtenidos por el componente OpenFileDialog
            calculaFechas();
        }
    }
}

```

```

private void calculaFechas()
{
    //Declaración de variables que se ocupan dentro del método calculaFechas()

    bool EsNumero = false;
    string line;
    string[] datos;
    string cadena = "";

    float Numero = 0;
    int conta = 0;
    System.IO.StreamReader file = new System.IO.StreamReader(DireccionText);

    //Se asigna la dirección a la caja de texto con el nombre: Seleccionar rango de fechas
    textBox1.Text = DireccionText;

    while ((line = file.ReadLine()) != null)
    {
        //lee cadena en el archivo de texto
        cadena = line;
        //separa todos los caracteres donde encuentre un espacio en blanco
        datos = cadena.Split(' ');
        //Valida si los valores que ha leído sean numéricos
        EsNumero = float.TryParse(datos[0].ToString(), out Numero);

        if (EsNumero == false)
        {
            for (int i = 0; i < datos.Length; i++)
            {
                if (datos[i] != "")
                {
                    if (conta == 0)
                    {
                        string[] Xs = datos[i].ToString().Split('-');
                        //En caso de que los valores encontrados no sean números, envía un mensaje de error al usuario.
                        if (Xs.Length != 3)
                        {
                            MessageBox.Show("Archivo incorrecto, imposible continuar.", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Error);
                            textBox1.Text = "";
                            return;
                        }
                    }
                    else
                    {
                        //Asigna en pantalla el rango de fechas que ha encontrado en el archivo
                        label3.Text = datos[i].ToString();
                    }
                }
                ListaInformacion.Add(datos[i]);
                conta++;
            }
        }
    }
    //Cierra el archivo de texto que ha leído de las fechas
    file.Close();

    ListaFechas = new string[ListaInformacion.Count]; //lista de fechas globales
    int Contarfechas=0;
    foreach (string element in ListaInformacion)
    {
        listBox1.Items.Add(element);
        ListaFechas[Contarfechas] = element;
        Contarfechas++;
    }
    label19.Text = (Contarfechas - 1).ToString(); //total de días

    listBox1.Items.Add("Se cargaron las fechas ");
    listBox1.Items.Add("Total de fechas " + label19.Text);
    label6.Text = ListaFechas[Contarfechas - 1].ToString();
}

private void CargaDatosDiarios()
{
    //Variables utilizadas dentro del método CargaDatosDiarios
    bool EsNumero = false;
    string line;
    string[] datos;
    string cadena = "";
    float Numero = 0;
    int conta = 0;

    //Lee el archivo de texto
    System.IO.StreamReader file = new System.IO.StreamReader(DireccionText);
    textBox2.Text = DireccionText;
    //Mientras que el archivo sea diferente de null, seguirá leyendo las líneas de texto
}

```

```
while ((line = file.ReadLine()) != null)
{
    cadena = line;
    //Separa cada línea si encuentra un espacio en blanco (TAB)
    datos = cadena.Split('\t');

    EsNumero = float.TryParse(datos[0].ToString(), out Numero);

    if (EsNumero == true)
    {
        for (int i = 0; i < datos.Length; i++)
        {
            if (datos[i] != "")
            {
                //Si es número el valor lo agrega dentro de la lista ListaValores que le corresponde a obtener los valores de la temperatura mínima
                ListaValores.Add(datos[i]);
                conta++;
            }
        }
    }
}
//Cierra el archivo
file.Close();

ListaTempMinima = new string[ListaValores.Count]; //lista de datos de las fechas globales
int ContarDatos = 0;

foreach (string element in ListaValores)
{
    //Agrega de la listaValores todos los datos a una nueva lista llamada ListaTempMinima
    ListaTempMinima[ContarDatos] = element;
    ContarDatos++;
}

label11.Text = (ContarDatos - 1).ToString(); //total de días
listBox1.Items.Add("Se cargaron todos los datos");
listBox1.Items.Add("Total de datos " + label11.Text);

int ContadorValor = 0;
string VLongitud = "";
string VLatitud = "";
string ValorN = "";
```

```

int PosicionFecha = 0;
string VFecha = "";
int TotaldeDias = int.Parse(label19.Text)+2;
for (int i = 0; i < ListaTempMinima.Length; i++)
{
    //Realiza el corrido de la lista para agregar la longitud, latitud y valor
    ContadorValor++;
    if (ContadorValor == 1) VLongitud = ListaTempMinima[i].ToString();
    if (ContadorValor == 2) VLatitud= ListaTempMinima[i].ToString();
    if (ContadorValor > 2)
    {
        VFecha=ListaFechas[PosicionFecha].ToString();
        ValorN = ListaTempMinima[i].ToString();
        PosicionFecha++;
        //Asigna al primer componente ListView la longitud, latitud y valor
        ListViewItem lvi = new ListViewItem(VFecha);
        lvi.SubItems.Add(VLongitud);
        lvi.SubItems.Add(VLatitud);
        lvi.SubItems.Add(ValorN);
        Lv.Items.Add(lvi);
    }

    if (ContadorValor>TotaldeDias)
    {
        i++;
        if (i < ListaTempMinima.Length)
        {
            VLongitud = ListaTempMinima[i].ToString();
            ContadorValor = 1; // empezar otro segmento
            PosicionFecha = 0;
        }
    }
}

private void CargaDatosDiariosMaxima()
{
    //Variables utilizadas dentro del método CargaDatosDiarios
    bool EsNumero = false;
    string line;
    string[] datos;
    string cadena = "";
float Numero = 0;
int conta = 0;

//Lee el archivo de texto
System.IO.StreamReader file = new System.IO.StreamReader(DireccionText);
textBox3.Text = DireccionText;
while ((line = file.ReadLine()) != null)
{
    cadena = line;
    //Separa cada línea si encuentra un espacio en blanco (TAB)
    datos = cadena.Split('\t');

    EsNumero = float.TryParse(datos[0].ToString(), out Numero);

    if (EsNumero == true)
    {
        for (int i = 0; i < datos.Length; i++)
        {
            if (datos[i] != "")
            {
                //Si es número el valor lo agrega dentro de la lista ListaValores que le corresponde a obtener los valores de la temperatura mínima
                ListaValoresMaxima.Add(datos[i]);
                conta++;
            }
        }
    }
}
//Cierra el archivo
file.Close();

ListaTempMaxima = new string[ListaValoresMaxima.Count]; //lista de datos de las fechas globales
int ContarDatos = 0;

foreach (string element in ListaValoresMaxima)
{
    //Agrega de la listaValores todos los datos a una nueva lista llamada ListaTempMinima
    ListaTempMaxima[ContarDatos] = element;
    ContarDatos++;
}

label11.Text = (ContarDatos - 1).ToString(); //total de días

```

```

listBox1.Items.Add("Se cargaron todos los datos");
listBox1.Items.Add("Total de datos maximo " + label11.Text);

int ContadorValor = 0;
string VLongitud = "";
string VLatitud = "";
string ValorN = "";
int PosicionFecha = 0;
string VFecha = "";
int TotaldeDias = int.Parse(label19.Text) + 2;
for (int i = 0; i < ListaTempMaxima.Length; i++)
{
    //Realiza el corrido de la lista para agregar la longitud, latitud y valor
    ContadorValor++;
    if (ContadorValor == 1) VLongitud = ListaTempMaxima[i].ToString();
    if (ContadorValor == 2) VLatitud = ListaTempMaxima[i].ToString();
    if (ContadorValor > 2)
    {
        VFecha = ListaFechas[PosicionFecha].ToString();
        ValorN = ListaTempMaxima[i].ToString();
        PosicionFecha++;
        //Asigna al primer componente ListView la longitud, latitud y valor
        ListViewItem lvi = new ListViewItem(VFecha);
        lvi.SubItems.Add(VLongitud);
        lvi.SubItems.Add(VLatitud);
        lvi.SubItems.Add(ValorN);
        Lv2.Items.Add(lvi);
    }

    if (ContadorValor > TotaldeDias)
    {
        i++;
        if (i < ListaTempMinima.Length)
        {
            VLongitud = ListaTempMaxima[i].ToString();
            ContadorValor = 1; // empezar otro segmento
            PosicionFecha = 0;
        }
    }
}
}

private void CargaDatosDiariosPrecipitacion()
{
    //Variables utilizadas dentro del método CargaDatosDiarios
    bool EsNumero = false;
    string line;
    string[] datos;
    string cadena = "";
    float Numero = 0;
    int conta = 0;

    //Lee el archivo de texto
    System.IO.StreamReader file = new System.IO.StreamReader(DireccionText);
    textBox4.Text = DireccionText;
    //Mientras que el archivo sea diferente de null, seguirá leyendo las líneas de texto
    while ((line = file.ReadLine()) != null)
    {
        cadena = line;
        //Separa cada línea si encuentra un espacio en blanco (TAB)
        datos = cadena.Split('\t');

        EsNumero = float.TryParse(datos[0].ToString(), out Numero);

        if (EsNumero == true)
        {
            for (int i = 0; i < datos.Length; i++)
            {
                if (datos[i] != "")
                {
                    //Si es número el valor lo agrega dentro de la lista ListaValores que le corresponde a obtener los valores de la temperatura mínima
                    ListaValoresPrecipitacion.Add(datos[i]);
                    conta++;
                }
            }
        }
    }
    //Cierra el archivo
    file.Close();

    ListaTempPrecipitacion = new string[ListaValoresPrecipitacion.Count]; //lista de datos de las fechas globales
    int ContarDatos = 0;

    foreach (string element in ListaValoresPrecipitacion)

```

```

{
    //Agrega de la listaValores todos los datos a una nueva lista llamada ListaTempMinima
    ListaTempPrecipitacion[ContarDatos] = element;
    ContarDatos++;
}

label11.Text = (ContarDatos - 1).ToString(); //total de dias
listBox1.Items.Add("Se cargaron todos los datos");
listBox1.Items.Add("Total de datos maximo " + label11.Text);

int ContadorValor = 0;
string VLongitud = "";
string VLatitud = "";
string ValorN = "";
int PosicionFecha = 0;
string VFecha = "";
int TotaldeDias = int.Parse(label9.Text) + 2;
for (int i = 0; i < ListaTempPrecipitacion.Length; i++)
{
    //Realiza el corrido de la lista para agregar la longitud, latitud y valor
    ContadorValor++;
    if (ContadorValor == 1) VLongitud = ListaTempPrecipitacion[i].ToString();
    if (ContadorValor == 2) VLatitud = ListaTempPrecipitacion[i].ToString();
    if (ContadorValor > 2)
    {
        VFecha = ListaFechas[PosicionFecha].ToString();
        ValorN = ListaTempPrecipitacion[i].ToString();
        PosicionFecha++;
        //Asigna al primer componente ListView la longitud, latitud y valor
        ListViewItem lvi = new ListViewItem(VFecha);
        lvi.SubItems.Add(VLongitud);
        lvi.SubItems.Add(VLatitud);
        lvi.SubItems.Add(ValorN);
        Lv3.Items.Add(lvi);
    }
}

if (ContadorValor > TotaldeDias)
{
    i++;
    if (i < ListaTempMinima.Length)
    {
        . . . . .
    }
}

```

```

        VLongitud = ListaTempPrecipitacion[i].ToString();
        ContadorValor = 1; // empezar otro segmento
        PosicionFecha = 0;
    }
}
}
}
private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text=="")
    {
        MessageBox.Show("Falta cargar las fechas ", "Verifique", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    DireccionText = TxtSimulacion.Open(openFileDialog1);

    if (DireccionText != "false")
    {
        //Botón para almacenar los datos de temperatura mínima
        CargaDatosDiarios();
    }
}

private void button3_Click(object sender, EventArgs e)
private void button4_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("Falta cargar las fechas ", "Verifique", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    DireccionText = TxtSimulacion.Open(openFileDialog1);

    if (DireccionText != "false")
    {
        //Botón para almacenar los datos de temperatura máxima
        CargaDatosDiariosMaxima();
    }
}

private void button5_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("Falta cargar las fechas ", "Verifique", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    DireccionText = TxtSimulacion.Open(openFileDialog1);

    if (DireccionText != "false")
    {
        //Botón para almacenar los datos de precipitación
        CargaDatosDiariosPrecipitacion();
    }
}

private void button3_Click_1(object sender, EventArgs e)
{
    //Valida que los campos tengas las rutas de los archivos que necesita para comenzar a generar los archivos
    if (Lv.Items.Count == 0)
    {
        MessageBox.Show("No existe informacion de temperatura mínima", "Error de usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    if (Lv2.Items.Count == 0)
    {
        MessageBox.Show("No existe informacion de temperatura máxima", "Error de usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    if (Lv3.Items.Count == 0)
    {
        MessageBox.Show("No existe informacion de precipitación", "Error de usuario", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
}

```

```

Cursor = Cursors.WaitCursor;
//Ejecuta el método GenerarArchivos que es el responsable de generar los archivos que Hydrotel necesita para operar
GenerarArchivos();
Cursor = Cursors.Default;
MessageBox.Show("Se ejecuto exitosamente la operación", "Aviso de interfaz", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}

public void GenerarArchivos()
{
    string fechainicial = label3.Text;
    string Fecha = "";
    string Maxima = "";
    string Minima = "";
    string Precipitacion = "";
    int contadorArchivo = 0;
    string NombreArchivo = "G_";
    string Cadena="";
    StreamWriter sw=null;
    //maxima
    int ContarDias = 0;
    int totalDias = int.Parse(label19.Text);

    //Selecciona un diectorio dentro de la raíz C:// y crea una carpeta llamada Meteo
    if (File.Exists(@"c:\Meteo") == false) Directory.CreateDirectory(@"c:\Meteo");

    for (int i = 0; i < Lv2.Items.Count; i++)
    {
        ContarDias++;
        Fecha = Lv2.Items[i].Text;
        if (ContarDias==1)
        {
            contadorArchivo++;

            Cadena = @"c:\Meteo\" + NombreArchivo + contadorArchivo.ToString() + ".met";
            if (File.Exists(Cadena))
            {
                Console.WriteLine("{0} already exists.", Cadena);
                return;
            }
            sw = File.CreateText(Cadena);

            Maxima = Lv2.Items[i].SubItems[3].Text;
            Minima= Lv2.Items[i].SubItems[3].Text;
            Precipitacion = Lv3.Items[i].SubItems[3].Text;

            //Escribe dentro del archivo meteo con la siguiente arquitectura
            sw.WriteLine(Fecha + " \t" + Maxima + " \t" + Minima + " \t" + Precipitacion);

            if (ContarDias > totalDias)
            {
                ContarDias = 0;
                sw.Close();
            }
        }
    }
}

private void button100_Click(object sender, EventArgs e)
{
    //Cierra ventana
    this.Close();
    this.Dispose();
}
}

```

Hidrograma.cs

La pantalla Hidrograma.cs contiene una conjunto de instrucciones de código para la lectura de datos en archivos de texto plano *.txt y además contiene una biblioteca de vínculo dinámico (DLL) para la graficar datos en forma de líneas senoidales, este componente se llama ZedGraph y se vincula al programa en tiempo de ejecución.

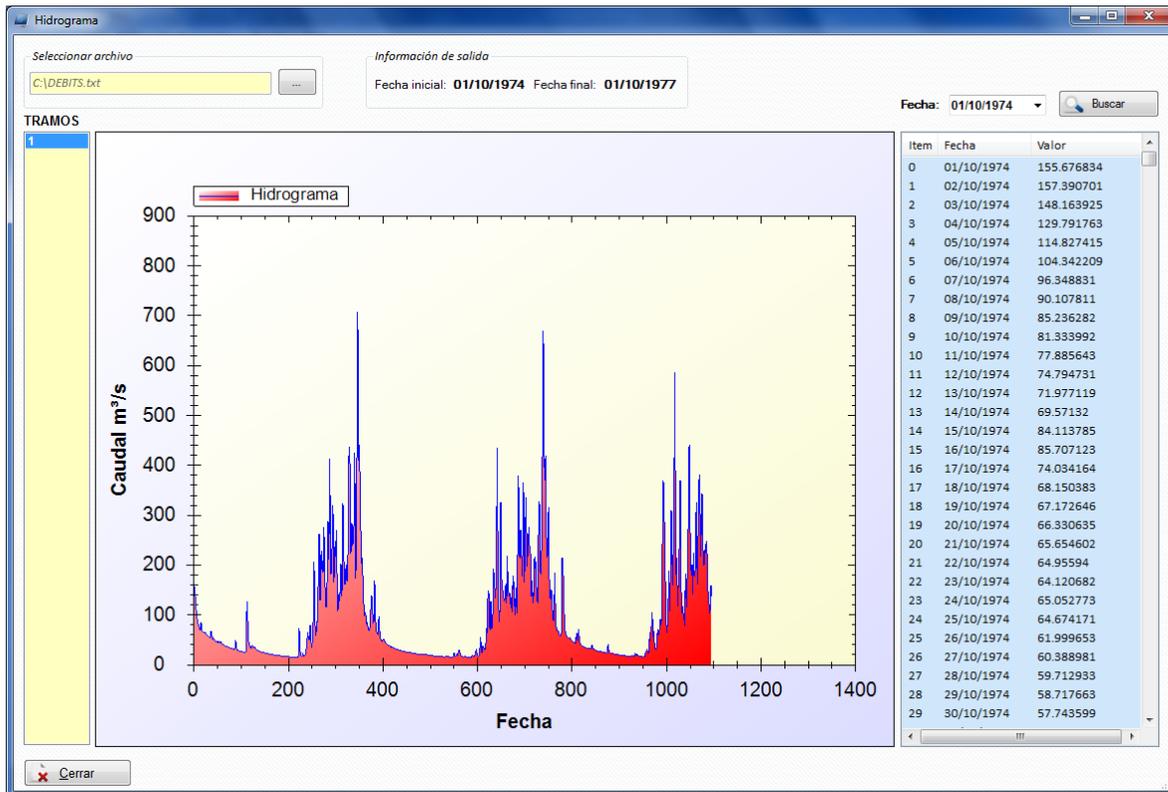


Figura 5. Pantalla Hidrograma

A continuación se muestra y describe el código fuente de la pantalla Hidrograma Figura 5. En estas líneas de código se observa las diversas operaciones dinámicas que el programa realiza para mostrar en pantalla toda una serie de datos en un solo click.

```
public partial class SacarTramos : Form
{
    //Declaración de variables por tipo de dato
    public string DireccionText = "";

    DateTime f1;
    DateTime f2;
    public List<double> ListaTramos = new List<double>();
    public List<double> ListaInformacion = new List<double>();

    public double[] valores;
    public string[,] MisFechas;

    public SacarTramos()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        //Limpiar listas
        ListaTramos.Clear();
        ListaInformacion.Clear();
        textBox1.Text = "";

        //Creación de un objeto GraphPane para graficar
        GraphPane myPane = zg1.GraphPane;
        zg1.GraphPane.CurveList.Clear();
        zg1.GraphPane.GraphObjList.Clear();

        //Asignar la dirección del archivo que contiene los datos de la precipitación

        listBox2.Items.Clear();
        DireccionText = TxtSimulacion.Open(openFileDialog1);

        if (DireccionText != "false")
        {
            //Si el componente OpenFileDialog contiene una ruta entonces limpiar y extraer los tramos
            //por medio del método calculaTramos()

            listView1.Items.Clear();
            listBox1.Items.Clear();
            listBox2.Items.Clear();
            textBox1.Text = DireccionText;

            //Método que calcula en número de tramos que contiene el archivo seleccionado
        }
    }
}
```

```

        calculaTramos();
    }
}

public void calculaTramos()
{
    //Variables utilizadas para realizar la extracción de los tramos contenidos en un archivo de texto.
    int counter = 0;
    int numeroTramos = 0;
    float Numero = 0;

    string line;
    string[] direccion;
    string[] datos;
    string cadena = "";

    bool EsNumero = false;

    string directorio = openFileDialog1.SafeFileName;

    direccion = directorio.Split('.');

    string nombre = direccion[0];
    bool EntroDebit = false; //Bandera que se utiliza para determinar si la instrucción ha encontrado la palabra 'DEBITS'

    // Abre y lee el archivo txt
    System.IO.StreamReader file = new System.IO.StreamReader(DireccionText);
    while ((line = file.ReadLine()) != null)
    {
        cadena = line;
        if (counter == 0) {
            //Valida que el archivo contenga números
            string [] datos1 = cadena.Split(' ');
            EsNumero = float.TryParse(datos1[0].ToString(), out Numero);
            if (EsNumero != true)
            {
                MessageBox.Show("Archivo incorrecto, por favor verifique.", "Error de selección de archivo", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }
        }
        //Estable el cursor en modo de espera
        Cursor = Cursors.WaitCursor;

        if (counter == 1) { f1 = DateTime.Parse(extraeFecha(counter, cadena)); label3.Text = f1.ToShortDateString(); label3.Visible = true; label5.Visible = false; }
        if (counter == 2)
        {
            f2 = DateTime.Parse(extraeFecha(counter, cadena)); label4.Text = f2.ToShortDateString(); label4.Visible = true;
        }
    }
}

```

```
        label6.Text = totalDias(f1, f2).ToString();
        label6.Visible = false;
    }

    if (counter == 3)
    {
        numeroTramos = int.Parse(line);
        if (numeroTramos != 1)
            numeroTramos = numeroTramos - 1;
    }

    if (counter > 3 && counter <= numeroTramos + 3)
    {
        listBox1.Items.Add("cargando el tramo " + cadena);
        ListaTramos.Add(double.Parse(cadena));
    }

    if (counter > (numeroTramos + 3))
    {
        datos = cadena.Split(' ');
        EsNumero = float.TryParse(datos[0].ToString(), out Numero);

        if (datos[0].ToString() == "DEBITS")
        {
            EntroDebit = true;
        }
        else
        {
            if (EsNumero == false) EntroDebit = false;
        }

        if (EsNumero == true)
        {
            if (EntroDebit == true)
            {
                for (int i = 0; i < datos.Length-1; i++)
                {
                    //Almacena los datos que se encuentran dentro de 'DEBITS'
                    ListaInformacion.Add(double.Parse(datos[i]));
                }
            }
        }
    }

    counter++;
```

```

}

file.Close(); //Cierra el archivo de texto
listBox1.Items.Add("Total de tramos: " + numeroTramos.ToString());
listBox1.Items.Add("Cargada la informacion DEBITS");

int xx=0;
valores = new double[Listainformacion.Count];
foreach (double element in ListaInformacion)
{
    valores[xx] = element;
    xx++;
}

listBox1.Items.Add("Total de valores:" + valores.Length.ToString());

foreach (double element in ListaTramos)
{
    double y = element;
    listBox2.Items.Add(y.ToString());
}

Cursor = Cursors.Default;
Mensaje muestra = new Mensaje();
muestra.ShowDialog();
}
//Método que ejecuta el hidrograma dependiendo del tramo seleccionado y envía los datos al componente ListView
public void SaberValores(int indice, double valortramo)
{
    int Tdias = int.Parse(label6.Text);
    int contador = indice;
    int PInicial = 0;
    int PFinal = 0;
    int resultado = 0;

    PFinal = (contador + 1) * Tdias;
    if (contador >= 1)
        PInicial = contador * Tdias;
    else
        PInicial = 0;

    resultado = PFinal - PInicial;
    listBox1.Items.Add("TRAMO NUMERO:" + valortramo.ToString());
    listBox1.Items.Add("Inicia " + PInicial + " Final " + PFinal + " Total Dias " + resultado.ToString());

    //Método para realizar el pintado de hidrograma dentro del componente Graph
    CreateGraph(zgl, PInicial, PFinal);
}

```

```

MandarFechas(Pinicial, PFinal);
listView1.Visible = true;
}

//Método que envía las fechas al componente listView
private void MandarFechas(int pinicia, int pfinal)
{
    int x = 0;

    listView1.Items.Clear();

    MisFechas = new string[3, valores.Length];
    DateTime FechaInicia = DateTime.Parse(label3.Text);
    dateTimePicker1.Value = FechaInicia;

    DateTime FechaFinal = DateTime.Parse(label4.Text);
    DateTime FechaRecorre = FechaInicia;
    double ValorFinal = 0;
    for (int i = pinicia; i < pfinal; i++)
    {
        string Fecha = FechaRecorre.ToString("dd/MM/yyyy");
        ValorFinal = valores[i];

        MisFechas[0, x] = x.ToString();
        MisFechas[1, x] = Fecha;
        MisFechas[2, x] = ValorFinal.ToString();

        ListViewItem lvi = new ListViewItem(MisFechas[0, x].ToString());
        lvi.SubItems.Add(MisFechas[1, x].ToString());
        lvi.SubItems.Add(MisFechas[2, x].ToString());
        listView1.Items.Add(lvi);

        FechaRecorre=FechaRecorre.AddDays(1);
        x++;
    }
}

//Método principal que recibe dos parametros que son los puntos uniciales y finales para realizar el hidrograma
private void CreateGraph(ZedGraphControl zgc, int pinicia, int pfinal)
{
    GraphPane myPane = zgc.GraphPane;
    zgc.GraphPane.CurveList.Clear();
    zgc.GraphPane.GraphObjList.Clear();

    // Estabeciendo títulos
    myPane.Title.Text = " ";
    myPane.XAxis.Title.Text = "Fecha";
    myPane.YAxis.Title.Text = "Caudal m³/s";

    // Establecer la función senoidal que será de puntos

```

```

PointPairList list = new PointPairList();
list.Clear();

int x = 0;
double ValorFinal = 0;
for (int i = pinicia; i < pfinal; i++)
{
    ValorFinal = valores[i];
    list.Add(x, ValorFinal);
    x++;
}

// Genera una curva azul con los simbolos redondos
LineItem myCurve = myPane.AddCurve("Hidrograma", list, Color.Blue, SymbolType.None);
// Pinta sobre la curva un area de color blanco-rojo con un gradiente de 45 grados
myCurve.Line.Fill = new Fill(Color.White, Color.Red, 45F);
// Pinta el simbolo de color blanco
myCurve.Symbol.Fill = new Fill(Color.White);

// Pinta los caracteres con un color degradado
myPane.Chart.Fill = new Fill(Color.White, Color.LightGoldenrodYellow, 45F);

// Pinta le panel de fondo con un gradiente
myPane.Fill = new Fill(Color.White, Color.FromArgb(220, 220, 255), 45F);

// Calculate the Axis Scale Ranges
zgc.AxisChange();

//Se deja algo de espacio extra en la parte superior de las etiquetas para que quepa en la gráfica
myPane.YAxis.Scale.MaxGrace = 0.2;

// Calcular la escala
zgc.IsShowPointvalues = true;
zgc.AxisChange();
zgc.Refresh();
zgc.RestoreScale(zgc.GraphPane);
zgc.Refresh();
}

//Método que extrae la fecha de un archivo de texto que contiene datos acerca de la precipitación
private string extraeFecha(int numLinea, string cadena)
{
    string anno = "", dia = "", mes = "";

    string[] fecha = cadena.Split('-');

```

```

    anno = fecha[0];
    mes = fecha[1];
    dia = fecha[2];

    string[] separaDosPunto = dia.Split(':');

    string retornaFecha = separaDosPunto[0] + "/" + mes + "/" + anno;

    return retornaFecha;
}

private double totalDias(DateTime f1, DateTime f2)
{
    //Método que calcula el total de días
    TimeSpan total = f2 - f1;
    double TotalDays = total.TotalDays;

    return TotalDays;
}

private void listBox2_DoubleClick(object sender, EventArgs e)
{
    //Evento que se dispara al realizar doble click
    int Indice = listBox2.SelectedIndex;
    double valor=double.Parse(listBox2.SelectedItem.ToString());
    SaberValores(Indice,valor);
}

private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    //Evento que se dispara al seleccionar un item del componente listView
    listView1.Visible = false;
    int Indice = listBox2.SelectedIndex;
    double valor = double.Parse(listBox2.SelectedItem.ToString());
    //Envia al método dos parametros que son el indice y value del tramo
    SaberValores(Indice, valor);
}

private void button2_Click(object sender, EventArgs e)
{
    bool Encontrado = false;
    listView1.Items.Clear();
    int totalDias = int.Parse(label6.Text);
    string FechaBuscar = dateTimePicker1.Value.ToString("dd/MM/yyyy");
    for (int i = 0; i < totalDias; i++)
    {
        if (MisFechas[1, i].ToString()==FechaBuscar)
        {
            //Llena el componente listView con el la fecha y el valor del tramo seleccionado.
            ListViewItem lvi = new ListViewItem(MisFechas[0, i].ToString());
            lvi.SubItems.Add(MisFechas[1, i].ToString());
            lvi.SubItems.Add(MisFechas[2, i].ToString());
            listView1.Items.Add(lvi);
            Encontrado = true;
            break;
        }
    }

    if (Encontrado == false)
    {
        MessageBox.Show("La fecha seleccionada no fue encontrada", "Verifique", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button100_Click(object sender, EventArgs e)
{
    //Cierra formulario
    this.Hide();
    this.Dispose();
}
}

```