



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

Desarrollo de un servicio web de simulación con  
aplicación en el diseño de sistemas de distribución de agua

Web service simulation development applied to the  
design of a water distribution network

Autor

Juan Carlos de Val Cremades

Directores

Asier Lacasta Soto

Pilar García Navarro

Escuela de Ingeniería y Arquitectura

2017



## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Juan Carlos de Val Cremades,

con nº de DNI 53766082V en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Grado de Ingeniería mecánica, (Título del Trabajo)

Desarrollo de un servicio web de simulación con aplicación en el diseño de  
sistemas de distribución de agua.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 26 de enero de 2017

Fdo: Juan Carlos de Val Cremades

# Desarrollo de un servicio web de simulación con aplicación en el diseño de sistemas de distribución de agua

## Resumen

El presente trabajo ha consistido en el desarrollo de una herramienta web al servicio de un simulador de redes de canales hidráulicos de lámina libre. Todo ello realizado en la nube, esto es, sin necesidad de descargar ni instalar software en el dispositivo conectado desde el que se esté utilizando. Es este punto, junto a la usabilidad, el que marca la diferencia entre este programa y algunos otros existentes que deben ser ejecutados en local desde su propio equipo.

La memoria mostrará, mediante un caso representativo, el proceso a seguir cuando se quiere simular una red de canales de este tipo así como los cálculos que la herramienta realiza para la obtención de los resultados así como las hipótesis en las que se basa.

Finalmente se representarán los datos obtenidos y se ilustrará la forma de interpretarlos con la herramienta desarrollada en el presente trabajo.

Escaneando el código QR de la parte inferior de esta página, o bien accediendo a la dirección web ([canalflowmodel.net/navigli/tfg\\_jcvc](http://canalflowmodel.net/navigli/tfg_jcvc)) se tendrá acceso tanto a esta memoria como a la herramienta descrita en el trabajo.



# Índice general

<b>Resumen</b>	<b>2</b>
<b>1. Introducción y alcance del trabajo</b>	<b>5</b>
<b>2. Modelo de simulación</b>	<b>7</b>
2.1. Formulación del flujo en el canal	7
2.2. Confluencias	10
2.3. Método de resolución	11
<b>3. Estructura de la herramienta</b>	<b>13</b>
<b>4. Caso de estudio</b>	<b>14</b>
<b>5. Diseño de la red</b>	<b>16</b>
5.1. Página de inicio	16
5.2. Elección número de tramos	16
5.3. Forma de las secciones	17
5.4. Topología de la red	18
5.5. Condiciones de contorno	20
5.6. Parámetros	22
<b>6. Resultados</b>	<b>25</b>
6.1. Gráfica de resultados en planta	25
6.2. Gráfica a lo largo de la longitud del tramo	27
6.3. Gráfica de una celda a lo largo de todo el tiempo de la simulación	29
6.4. Tique y edición de casos	30
6.5. Validación de resultados en Navigli	31
<b>7. Atajos de Navigli</b>	<b>33</b>
7.1. Edición de casos	33
7.2. Visualización rápida de resultados	34
<b>8. Conclusiones</b>	<b>35</b>
<b>Bibliografía</b>	<b>36</b>
<b>Anexos</b>	<b>37</b>
Anexo 1. Validación caso estacionario	37
Anexo 3. Código fuente del programa	53

# Índice de figuras y tablas

Figura 1. Sección transversal trapecio.	8
Figura 2. Sección transversal triángulo.	8
Figura 3. Sección transversal semicírculo.	9
Figura 4. Confluencia ejemplo de tres canales.	11
Figura 5. Figura explicativa CFL.	12
Figura 6. Estructura de la herramienta web.	13
Figura 7. Gráfica en planta de la red de canales de ejemplo.	14
Tabla 1. Datos geométricos de la red a simular.	14
Tabla 2. Condiciones de contorno de la red a simular.	15
Figura 8. Pantalla de inicio.	16
Figura 9. Pantalla para la elección el número de tramos y nombre del caso.	17
Figura 10. Selección de forma de las secciones de cada tramo y Manning.	18
Figura 11. Información página topología.	19
Figura 12. Página de topología.	19
Figura 13. Configuración avanzada de cada tramo.	20
Figura 14. Condiciones de contorno de la entrada a la red.	21
Figura 15. Condiciones de contorno de salida de la red.	22
Figura 16. Parámetros de la simulación.	22
Figura 17. Ejemplo mensaje denegando la generación de más de 100 instantes de simulación por tramo.	23
Figura 18. Evolución gráfica gradiente en planta.	25
Figura 19. Gráfica en planta observando el número de Froude.	26
Figura 20. Cuadro de diálogo para modificar el rango de color	27
Figura 21. Gráfica de la red con la escala introducida manualmente.	27
Figura 22. Gráfica a lo largo del tramo.	28
Figura 23. Evolución variable $z+h$ mostrada en la gráfica a lo largo del tramo.	29
Figura 24. Gráfica (Figura 22) de tramo con el zum en un rango concreto.	29
Figura 25. Gráfica de celda a lo largo del tiempo de la simulación.	30
Figura 26. Pie de página de resultados donde se puede comenzar la edición.	30
Figura 27. Aumento del calado en la celda 11 del primer canal de 0,6 m aproximadamente.	31
Figura 28. Aumento del calado en la celda 12 del primer canal de 0,6 m aproximadamente.	31
Figura 29. Caudal de la celda 11 del primer canal.	32
Figura 30. Caudal de la celda 12 del primer canal.	32
Figura 31. Edición de un caso con el mismo nombre que el original.	33
Figura 32. Ventana para la vista rápida de casos generados.	34

# 1. Introducción y alcance del trabajo

La herramienta se enmarca en un desarrollo anterior existente llamado CanalFlowModel<sup>[1]</sup> del Grupo de Hidráulica Computacional (ghc.unizar.es) en el marco del proyecto BIA2011-20192-C02-01: GENERACIÓN DE UNA HERRAMIENTA INFORMÁTICA DE CONTROL ÓPTIMO DE COMPUERTAS EN REDES DE CANALES PARA SISTEMAS DE RIEGO financiado por el Ministerio de Economía y Competitividad.

En esta herramienta se trata de extender la anterior a la simulación de redes de canales.

Mediante el uso de la herramienta desarrollada en este trabajo, será posible realizar el diseño de redes de canales que sirvan para acercar los recursos de agua a las poblaciones que lo demandan. De esta manera, este trabajo pretende ayudar del lado del desarrollo tecnológico ya que esta herramienta ingenieril permite realizar simulaciones y observar resultados de una manera rápida y sencilla.

A la herramienta creada se le ha puesto el nombre *Navigli* que significa Canales en italiano. Éste es el nombre de uno de los barrios más modernos y cosmopolitas de Milán (Italia) ciudad con una amplia experiencia en cuanto a canales artificiales se refiere. *Navigli* es una herramienta novedosa ya que, siendo cierto que existen otros programas de cálculo, como HEC-RAS<sup>[2]</sup> o MIKE11<sup>[3]</sup>, software libre y comercial, respectivamente. *Navigli* está desarrollado en la nube, es decir, cualquier dispositivo con conexión a internet es capaz de simular una red de canales sin necesidad de descargar ni instalar software adicional. Tiene también la ventaja de que los casos generados se guardan en la nube y el usuario puede acceder fácilmente a los resultados o editarlos sin necesidad de guardar nada más que el nombre que su caso tenga.

Durante los siguientes capítulos se planteará un caso ficticio, tomado de dos artículos<sup>[4][5]</sup>, en el cual se puedan observar un número de variables significativas para poder ser consciente de las posibilidades que ofrece la herramienta desarrollada.

En la herramienta desarrollada se han trabajado los siguientes aspectos:

- Planificación de los pasos a seguir para conseguir diseñar la red deseada por el usuario en un orden que facilite la comprensión del propio proceso de diseño pero que, a su vez, demande la introducción de datos en su momento indicado. El orden es primordial para no caer en casos esquina o incoherencias.
- Diseño gráfico de una experiencia de usuario positiva generando una interfaz amigable, limpia y clara.
- Desarrollo mediante tecnologías de programación web como PHP para la parte del servidor, HTML en lo relacionado al diseño web, CSS en el diseño gráfico del documento HTML, JavaScript y jQuery se han utilizado para la interacción entre el usuario y la página y AJAX para la obtención de los resultados del servidor. La página web se estructura en cinco formularios que van enviándose datos de uno a otro mientras van haciendo cálculos hasta llegar a las últimas pantallas donde se escriben los archivos necesarios para la interpretación del modelo de simulación. Se ha utilizado algo de código abierto para el apoyo a nivel visual y gráfico como son los frameworks Twitter Bootstrap, Chroma js,

Google Fonts y Highcharts que permiten utilizar partes de código ya desarrollado y adaptarlo a las necesidades de cada programa.

- Vinculación con las herramientas de cálculo ya existentes en el Grupo de Hidráulica Computacional.

La estructura de la memoria es como sigue:

- En el capítulo 2 se explica el modelo de flujo y de simulación.
- El capítulo 3 introducirá el flujo que se sigue a la hora de utilizar la herramienta así como sus caminos alternativos.
- El caso mediante el cual se va a mostrar el funcionamiento de Navigli se mostrará en el capítulo 4.
- El diseño geométrico de la red y sus propiedades se verá en el capítulo 5.
- En el capítulo 6 se mostrarán las gráficas de resultados, su manejo, manipulación e interpretación así como la verificación de la herramienta.
- El capítulo 7 servirá como explicación de los caminos alternativos o atajos para el ágil uso Navigli.
- Finalmente se cerrará, en el capítulo 8, con las conclusiones obtenidas del desarrollo de esta herramienta en el marco del Trabajo de Fin de Grado.

Se añadirá una validación de un caso estacionario, un manual de usuario y el código realizado en los anexos.

Durante la memoria se hará referencia a algunas líneas de código fuente en las que se desarrolla lo referenciado.

## 2. Modelo de simulación

El cálculo lo realiza un modelo de simulación numérica desarrollado en el área de Mecánica de Fluidos, para flujo de lámina libre. El modelo se basa en las siguientes hipótesis<sup>[6][7]</sup>:

- El flujo se supone incompresible.
- El flujo es unidimensional y la velocidad uniforme en la sección transversal.
- Se aplicará un modelo de flujo estacionario (Manning) para considerar las pérdidas energéticas por fricción.
- Se asume un modelo de presiones hidrostáticas.
- La pendiente media del fondo  $\theta$  es pequeña, por lo que se pueden aplicar las siguientes aproximaciones:  $\cos\theta \sim 1$ ,  $\sin\theta \sim \tan\theta \sim \theta$ .

El modelo distingue la formulación dentro de un canal recto de la de las confluencias.

### 2.1. Formulación del flujo en el canal

En general, se acepta que un flujo transitorio de flujos en superficie libre, en una aproximación unidimensional puede ser formulado por las ecuaciones de Saint-Venant. Estas representan la conservación de masa y movimiento a lo largo de la dirección del flujo. Mediante estas ecuaciones se puede modelar la mayoría de los problemas asociados al flujo en canales:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad (1)$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \frac{Q^2}{A} + gI_1 \right) = gI_2 + gA(S_0 - S_f) \quad (2)$$

Estas ecuaciones representan la conservación de masa y cantidad de movimiento respectivamente, siendo  $A$  el área mojada de la sección transversal,  $Q$  el caudal y  $g$  la aceleración debida a la gravedad.  $I_1$  representa el término de la fuerza de presión hidrostática sobre las secciones transversales.

$$I_1 = \int_0^h (h - \eta)b(\eta)d\eta \quad (3)$$

con: 
$$b(\eta) = \frac{\partial A(\eta)}{\partial \eta} \quad (4)$$

La variable  $\eta$  corresponde a la altura de un punto desde el fondo de la sección,  $h$  es la profundidad máxima o calado en cada sección transversal,  $I_2$  es el término de fuerzas de presión debido a la variación del ancho del canal:

$$I_2 = \int_0^{h(x,t)} (h - \eta) \frac{\partial b(x,\eta)}{\partial x} d\eta \quad (5)$$

Como todos los casos que se van a estudiar en el presente trabajo son con canales en formas prismáticas, el término  $I_2$  de presión no se utilizará.

En la parte derecha de la ecuación (2) se encuentran los términos fuente provenientes de la pendiente del fondo y las pérdidas por fricción. La pendiente del fondo del canal se define como:

$$S_0 = -\frac{\partial z}{\partial x} \quad (6)$$

Donde  $z$  es la cota del fondo del canal.

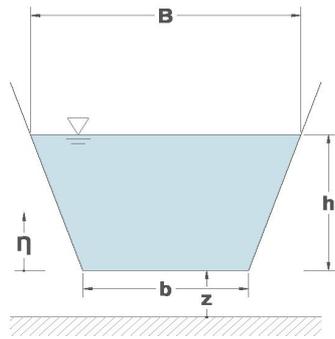
Las pérdidas por fricción dado por el coeficiente  $n$  de Manning:

$$S_f = \frac{Q|Q|n^2}{A^2 R_H^{4/3}} \quad (7)$$

con radio hidráulico  $R_H = A/P$ ,  $P$  es el perímetro mojado.

Dependiendo de las formas de las secciones que se pueden dar en nuestros problemas debemos distinguir:

- Rectangular o trapezoidal:

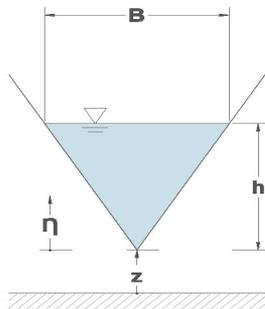


$$P = B + b + \sqrt{(B - b)^2 + 4h^2} \quad (8)$$

$$A = \frac{1}{2}(B + b)h \quad (9)$$

Figura 1. Sección transversal trapecio.

- Triangular:

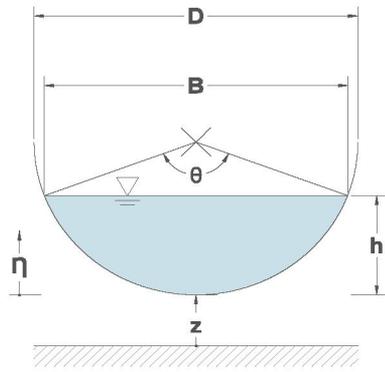


$$P = 2\sqrt{\frac{b^2}{4} + h^2} + B \quad (10)$$

$$A = \frac{1}{2}Bh \quad (11)$$

Figura 2. Sección transversal triángulo.

- Semicircular:



$$P = \frac{\theta D}{2} \quad (12)$$

$$A = \frac{(\theta - \sin\theta)D^2}{8} \quad (13)$$

Figura 3. Sección transversal semicirculo.

Siendo:

- $P$ : Perímetro mojado de la sección transversal al canal.
- $A$ : Superficie mojada de la sección transversal al canal.
- $B$ : Ancho de la superficie libre del agua.
- $b$ : Ancho del fondo del canal.
- $h$ : Altura o calado del agua.
- $R$ : Radio del semicírculo.

La celeridad de las ondas superficiales  $c$  está definida como:

$$c^2 = \frac{\partial I_1}{\partial A} = g \frac{A}{B} \quad (14)$$

Y según su forma:

- Rectangular/trapezoidal:

$$c^2 = g \frac{1}{2} \frac{(B+b)h}{B} \quad (15)$$

- Triangular:

$$c^2 = g \frac{h}{2} \quad (16)$$

- Semicircular:

$$c^2 = \frac{(\theta - \sin\theta)D^2}{8B} \quad (17)$$

El sistema (ecuaciones 1 y 2) puede formularse en forma vectorial:

$$\frac{\partial \bar{U}}{\partial t} + \frac{\partial \bar{F}}{\partial x} = \bar{R} \quad (18)$$

En esta expresión  $\bar{U}$ ,  $\bar{F}$  y  $\bar{R}$  representan las variables conservadas, los flujos y los términos fuente, respectivamente:

$$\bar{U} = (A, Q)^T \quad (19)$$

$$\bar{F} = (Q, \frac{Q^2}{A} + gI_1)^T \quad (20)$$

$$\bar{R} = (0, gA(S_0 - S_f))^T \quad (21)$$

La matriz Jacobiana del sistema es:

$$\bar{J} = \frac{\partial \bar{F}}{\partial \bar{U}} = \begin{pmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{pmatrix} \quad (22)$$

Donde  $u = \frac{Q}{A}$ , siendo  $u$  la velocidad (promedio en la sección) del agua.

Los autovalores y autovectores del sistema son los siguientes:

$$\bar{\lambda}^{1,2} = u \pm c, \quad \bar{e}^{1,2} = (1, u \pm c)^T \quad (23)$$

Como la matriz jacobiana posee dos autovalores reales y es diagonalizable (sus autovectores son linealmente independientes), se dice que el sistema pertenece a la familia de ecuaciones hiperbólicas y por tanto la estabilidad numérica la controlan los valores propios.

En este tipo de problemas resulta común la caracterización del tipo de flujo a través del número de Froude (análogo al número de Mach en flujo compresible). Se trata de una magnitud adimensional definida por:

$$Fr = \frac{u}{c} \quad (24)$$

Si el número de Froude es mayor a 1 se dice que el flujo es supercrítico, en el caso igual a 1 crítico y cuando el número es inferior a 1 el fluido está en estado subcrítico.

## 2.2. Confluencias

Mientras el flujo a lo largo del canal se considera unidireccional, en las uniones o confluencias el movimiento es más complejo. Para generar una solución rápida, se va a utilizar una solución basada en balances locales dentro de la confluencia. Constituyen las condiciones de contorno de los tramos que allí se unen. Usaremos un ejemplo para ilustrarlo, en él se supone un flujo desde 1 hasta 2 y 3.

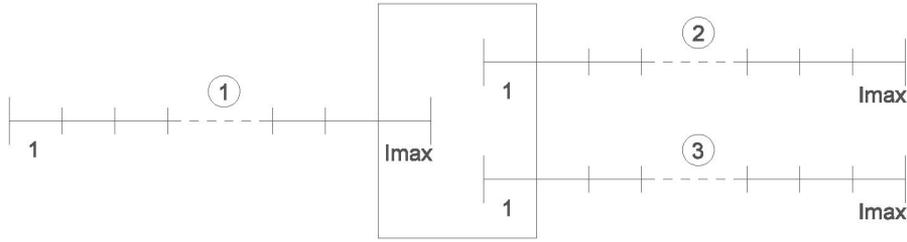


Figura 4. Confluencia ejemplo de tres canales.

Por un lado, se exigirá que el nivel superficial a la entrada de las tuberías 2 y 3 sea igual a la salida de la tubería 1:

$$(h + z)_2(1) = (h + z)_3(1) = (h + z)_1(l_{max}) \quad (25)$$

En cuanto al caudal, se exige continuidad.

$$(h + z)_1 = (h + z)_2 = \dots = (h + z)_N \quad \sum_{i=1}^N Q_i = 0 \quad (26)$$

### 2.3. Método de resolución

El método numérico empleado se basa en el esquema de Roe de primer orden, enmarcado dentro de la familia de los métodos de volúmenes finitos. Se trata de un método adaptado a las situaciones transitorias, capaz de trabajar en régimen, subcrítico, mixto y supercrítico. Es un método de resolución explícito y por ello el tamaño de paso de tiempo está acotado por la condición de estabilidad CFL.

En general, un método numérico se considera estable si las perturbaciones de la solución se mantienen acotadas. De lo contrario, el error en la solución crece de forma exponencial y la calidad de los resultados numéricos se ve seriamente comprometida. Una de las principales causas de las inestabilidades numéricas es el hecho de que la región de influencia física. En los esquemas explícitos, como el empleado en este trabajo, la región de influencia numérica viene determinada por el tamaño de celda  $\Delta x$ , ya que el valor de una variable depende de los valores de las celdas contiguas. Por otro lado, la región de influencia física viene dada por la distancia a la cual se ha podido propagar la información a velocidad  $c$ , es decir,  $(|u| \pm c)\Delta t$ . Por lo tanto, una primera forma de escribir la condición de estabilidad es:

$$\Delta x \geq (|u| \pm c)\Delta t \quad (27)$$

De la expresión anterior, se puede deducir que el tamaño máximo para el paso temporal es:

$$\Delta t_{max} = \frac{\Delta x}{\max(|u| \pm c)} \quad (28)$$

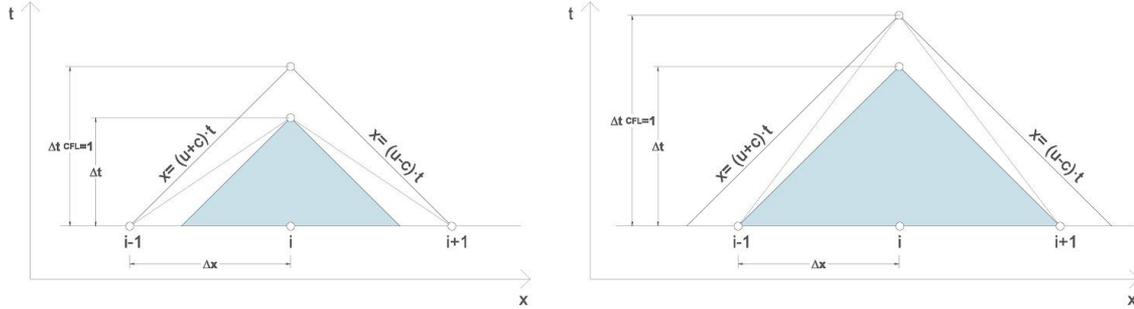


Figura 5. Figura explicativa CFL.

Definiendo el número de Courant-Friedrichs-Lewy (CFL), utilizado en la herramienta que este trabajo explica, como:

$$CFL = \frac{\Delta t}{\Delta t_{max}} < 1 \quad (29)$$

Se obtiene que la condición de estabilidad para un esquema explícito consiste en mantener este coeficiente por debajo de la unidad.

En el modelo permite la utilización de un paso de tiempo dinámico sin más que fijar el CFL deseado. En cada paso se recalcula internamente el  $\Delta t$  en función de la malla  $\Delta x$  y del flujo.

$$\Delta t = CFL \cdot \frac{\Delta x}{\max(|u| \pm c)} \quad (30)$$

### 3. Estructura de la herramienta

El objetivo principal de este trabajo ha sido el desarrollar una herramienta fácil de utilizar por los usuarios que lo precisen siendo lo más versátil e intuitiva posible. Por ello se plantea la siguiente estructura de la herramienta en la figura 6:

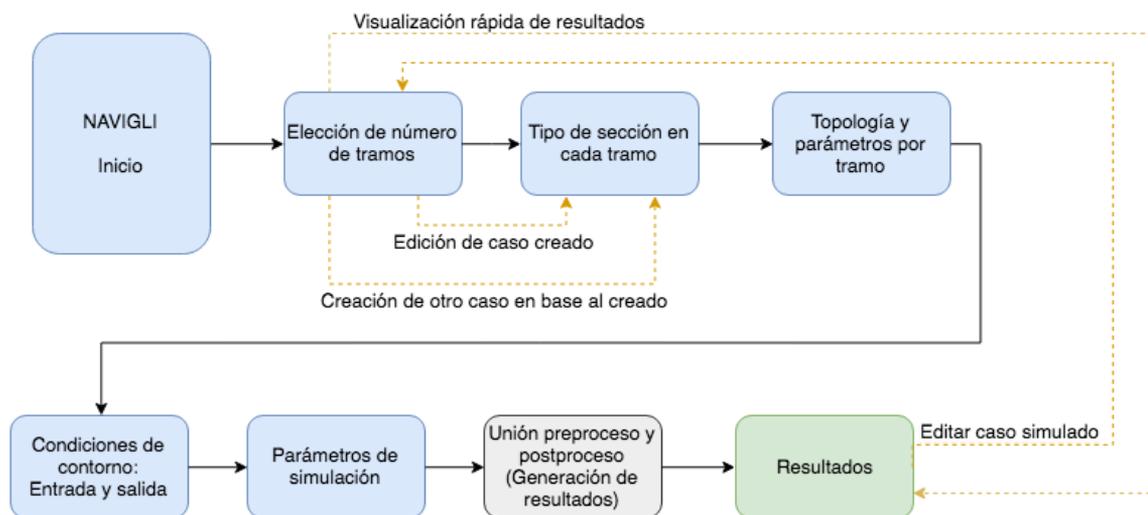


Figura 6. Estructura de la herramienta web.

Debido a que la herramienta diseña redes de canales y éstas pueden tener un número  $N$  de canales, se ha decidido dividir el proceso en varios pasos ya que si se hiciese en una sola página su longitud podría aumentar con el incremento de tramos que componen la red. Esta cuestión mejora la usabilidad de la parte del preproceso de la aplicación.

Existen una serie de atajos creados en *Navigli* con el fin de agilizar la visualización y edición de casos generados anteriormente.

## 4. Caso de estudio

A la vez que se va a explicar en este trabajo el funcionamiento de la herramienta desarrollada, se ilustrará mediante un caso de estudio. Este caso es planteado en distintas publicaciones<sup>[4][5]</sup> en las que se ilustran tanto la definición de la red como los resultados obtenidos.

Siendo la gráfica en planta la mostrada en la figura 7.

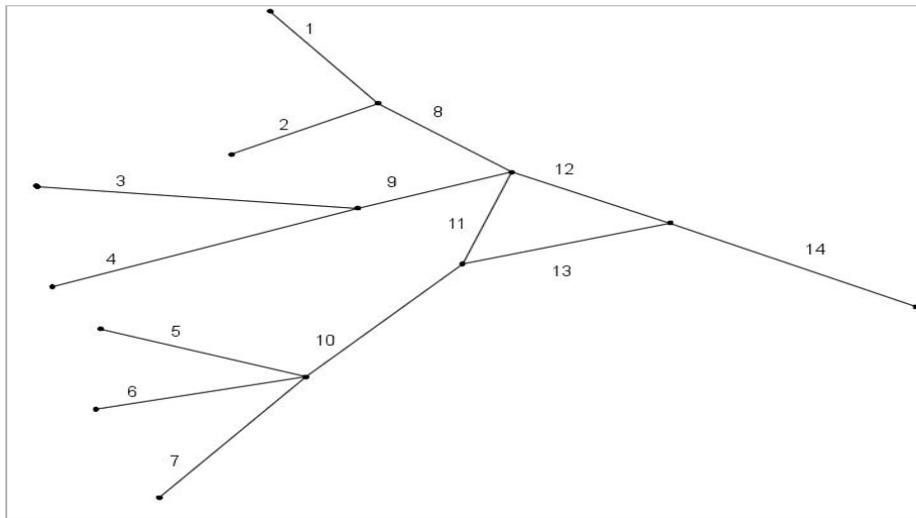


Figura 7. Gráfica en planta de la red de canales de ejemplo.

Los datos geométricos y de condiciones de contorno correspondientes a la red ilustrada en la figura 7 son los de las tablas 1 y 2, respectivamente.

Tramo	Longitud	Pendiente	Manning	Núm. celdas	Tipo	Ancho del fondo	Pendiente lateral
1	1500	0,00027	0,022	15	Trapezoidal	10	1
2	1500	0,00027	0,022	15	Trapezoidal	10	1
3	3000	0,00047	0,025	30	Trapezoidal	10	1
4	3000	0,00047	0,025	30	Trapezoidal	10	1
5	2000	0,0003	0,022	20	Trapezoidal	10	1
6	2000	0,0003	0,022	20	Trapezoidal	10	1
7	2000	0,0003	0,022	20	Trapezoidal	10	1
8	1500	0,00027	0,022	15	Trapezoidal	10	1
9	1500	0,00027	0,022	15	Trapezoidal	10	1
10	2000	0,0003	0,022	20	Trapezoidal	10	1
11	1200	0,00025	0,022	12	Rectangular	10	Vertical
12	1600	0,0005	0,022	36	Rectangular	20	Vertical
13	2000	0,00025	0,022	20	Rectangular	20	Vertical
14	2500	0,00016	0,022	25	Rectangular	20	Vertical

Tabla 1. Datos geométricos de la red a simular.

Entradas 1, 2, 3, 4, 5, 6 y 7		Salida 14	
t [s]	Q [m <sup>3</sup> /s]	t [s]	z+h [m]
0	10	0	2,5
143200	10	143200	2,5
172000	15	172000	2,9
200800	10	200800	2,5
244000	10	244000	2,5

Tabla 2. Condiciones de contorno de la red a simular.

Las entradas de agua en la red son en los tramos 1, 2, 3, 4, 5, 6 y 7 y la salida es única por el canal número 14.

En esta red se toman una serie de consideraciones previas:

- La red parte con un calado inicial de 0,2 metros.
- En el instante inicial de la simulación el caudal  $Q$  es igual a 0 m<sup>3</sup>/s.
- La simulación se realiza con agua.
- El calado mínimo en cada tramo es igual a 0,001 metros.

En este caso se aplicará un caudal constante hasta un tiempo aproximado de 39 horas con el fin de que la red esté estabilizada antes de introducir la variación de caudal que hará de este ejemplo un caso transitorio.

Finalmente se comprobará si los resultados obtenidos mediante esta herramienta se corresponden a los expuestos en las publicaciones mencionadas.

## 5. Diseño de la red

### 5.1. Página de inicio

En primer lugar, como se ilustra en la figura 8, se muestra una página inicio donde hay, pulsando el botón de información [Anexo3, Página de inicio, HTML línea 39], una breve explicación de la herramienta así como un enlace a su versión en inglés, un manual de usuario y un vínculo a la web [Anexo3, Página de inicio, HTML línea 62] de EINA (Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza). En esta memoria recogeremos el proceso en idioma español, no obstante la versión inglesa es análoga. Para comenzar con el proceso de diseño se debe presionar el botón Comenzar [Anexo3, Página de inicio, HTML línea 78] .



Figura 8. Pantalla de inicio.

### 5.2. Elección número de tramos

En el siguiente paso se define el número de tramos [Anexo3, Elección de tramos, HTML línea 92] y se elige un nombre al caso [Anexo3, Elección de tramos, HTML línea 109] que el usuario va a crear desde ese momento como se puede ver en la figura 9. Más tarde al nombre del caso se le añadirá un identificador único para que dos casos con el mismo nombre no se confundan entre sí.

NUMERO DE TRAMOS

NAVIGLI

¿Tienes ya un caso simulado?

Número de tramos de la red 14

Pon nombre a tu simulación WILEY\_TRANSITORIO

Siguiente →

Figura 9. Pantalla para la elección el número de tramos y nombre del caso.

Nótese también la existencia de un botón con el texto *¿Tienes ya un caso simulado?* [Anexo3, Elección de tramos, HTML línea 37]. Mediante este botón será posible abrir un caso simulado anteriormente y poder editarlo. Se explicará en el capítulo 7 *Atajos de Navigli*.

Una vez el usuario ha introducido los datos necesarios debe pulsar *Siguiente* [Anexo3, Elección de tramos, HTML línea 123].

### 5.3. Forma de las secciones

La herramienta Navigli accede al siguiente paso, mostrado en la figura 10, en el cual se define la forma de la sección de cada tramo supuesta constante a lo largo de toda su longitud. Se proponen tres tipos [Anexo3, Elección de tramos, HTML línea 78] de formas: semicircular, triangular y trapezoidal (pudiendo ser rectangular o cuadrada con las dimensiones adecuadas).

Todos los tramos se encuentran en la misma página pero solamente se verá el que el usuario esté editando [Anexo3, Forma de las secciones, Javascript para la paginación, línea 3], de esta manera no llenamos la pantalla con datos que no estamos editando en ese momento, con el fin de no confundir al usuario.

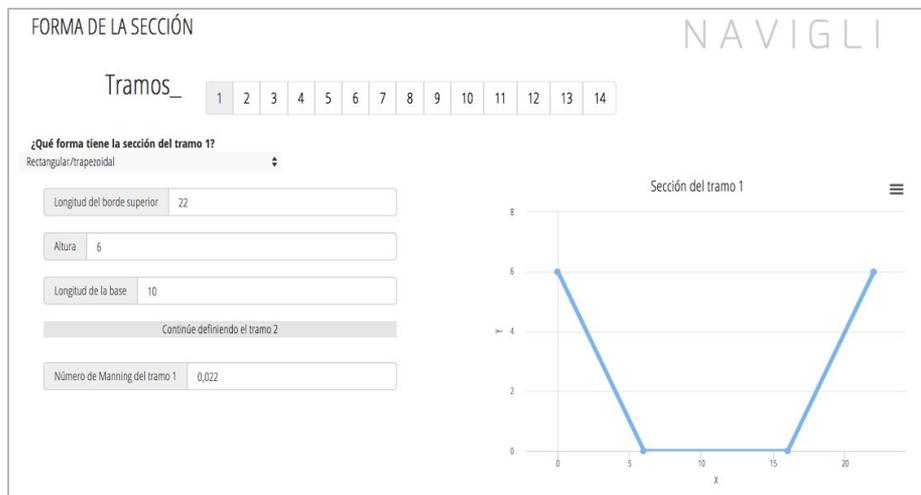


Figura 10. Selección de forma de las secciones de cada tramo y Manning.

En esta parte se define, también por tramo, el coeficiente de Manning [Anexo3, Forma de las secciones, HTML línea 183] que se utiliza para el cálculo de las pérdidas por la fricción. En el caso que presentamos en este documento todos los tramos son rectangulares/trapezoidales y sus datos ya han sido expuestos anteriormente.

Una vez se han rellenado todos los datos relacionados con la sección para todos los tramos que conforman la red aparece el botón *Siguiente* [Anexo3, Forma de las secciones, HTML línea 201] para avanzar a la parte de diseño topológico de la red. Solo aparece el botón en el último tramo para evitar confusiones y que, de esta manera, se presione solamente después de haber pasado por todos los tramos rellenando los datos correspondientes.

## 5.4. Topología de la red

En el siguiente paso, se define las propiedades topológicas de la red. Sólo es necesario introducir coordenadas en  $x$  e  $y$  (planta de la red) y cota absoluta en  $z$  para la entrada y la salida. Para comprender mejor a qué puntos corresponden estas entradas y salidas existe a disposición del usuario un botón de información en la esquina superior derecha [Anexo3, Topología, HTML línea 55]. Esto se ilustra en la figura 11.

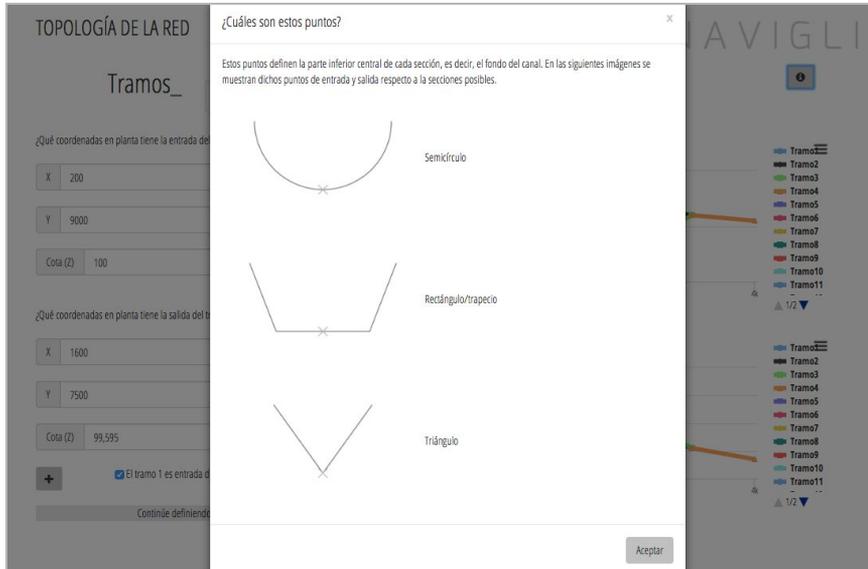


Figura 11. Información página topología.

El usuario conformará la red de canales tanto en planta como en alzado mediante las entradas y salidas de los distintos tramos. El usuario deberá marcar también los tramos que son entrada de agua en la red mediante una casilla donde se pregunta si el tramo es entrada de la red [Anexo3, Topología, HTML línea 243]. En el caso que se está simulando se han marcado como entrada los tramos 1, 2, 3, 4, 5, 6 y 7.

Se trata de distinguir lo mejor posible los tramos entre sí y por ello en esta pantalla se adjudica un color diferenciador a cada tramo, ilustrado en la figura 12.

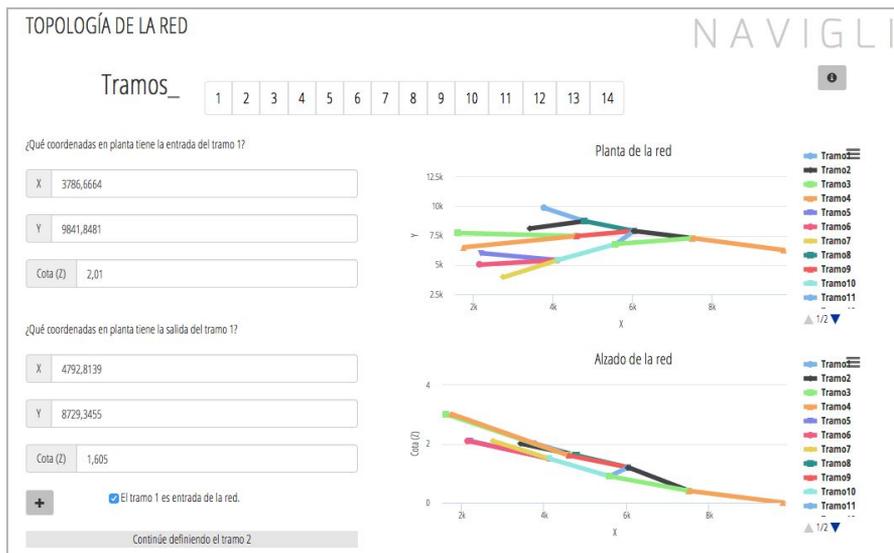


Figura 12. Página de topología.

Junto a la marca donde pregunta al usuario si se trata de una entrada de la red se encuentra un botón con el símbolo “+” [Anexo3, Topología, HTML línea 240] para una configuración más

avanzada de cada tramo. Si no se desea realizar esta configuración estos valores vendrán por defecto con el valor mostrado en la figura 13.

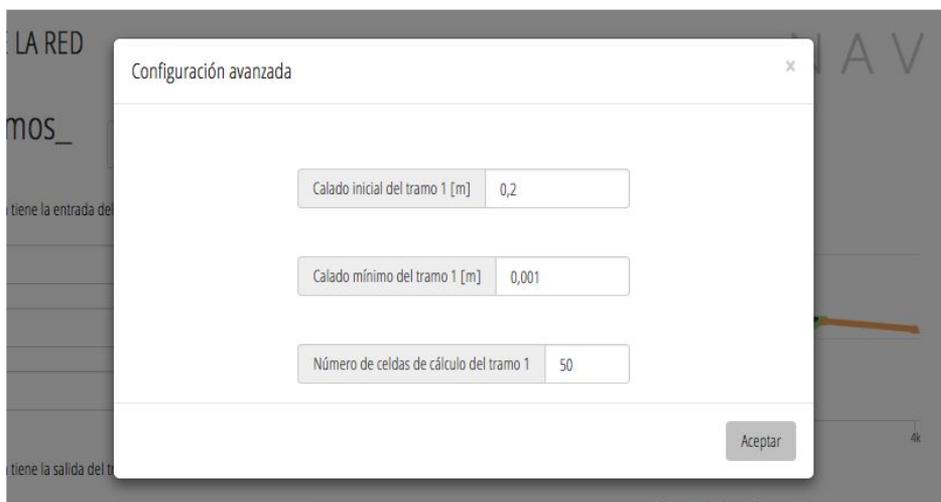


Figura 13. Configuración avanzada de cada tramo.

En esta vista hay tres datos a introducir:

- El calado inicial, es el calado con el que parte la red antes de iniciar la simulación.
- El calado mínimo de agua en el tramo en cuestión y se utiliza para anular la velocidad en caso de que el calado sea inferior al mínimo definido.
- El número de celdas de cálculo de cada tramo.

Una vez rellenados estos datos (en el caso que se desee una configuración avanzada) se deberá pulsar o bien el botón *Aceptar* o bien en cualquier lugar sombreado que ha quedado en segundo plano. En este caso se dejarán todos los valores por defecto.

Como anteriormente, una vez completados todos los datos y en la vista del último tramo (si no aparecerá el letrero visto en la figura 12 *Continúe definiendo el tramo 2 [Anexo3, Topología, HTML línea 333]*) se deberá pulsar el botón *Siguiente* para continuar definiendo más parámetros de simulación.

Cuando el usuario pulsa *Siguiente*, todos los datos geométricos necesarios se calculan internamente [*Anexo3, Condiciones de contorno, HTML línea desde 147 hasta 344*] situando las secciones en la posición perpendicular a los tramos formados por el segmento de entrada y salida.

## 5.5. Condiciones de contorno

En el siguiente paso se definirán las condiciones de contorno de las entradas y las salidas de la red diseñada.

La herramienta está programada para que automáticamente se conozca qué tramos tienen uniones y qué tramos son salida de la red. Lo realiza [*Anexo3, Condiciones de contorno, HTML línea desde 537 hasta 654*] comparando las coordenadas  $x$  e  $y$  de los extremos de todos los tramos de la red y viendo si coinciden. Las coordenadas  $z$  no se tienen en cuenta en esta comparación ya

que *Navigli* está preparado para asumir escalones en el fondo de la red de canales. De esta manera si un punto en un tramo no es entrada y tiene una unión o no tiene uniones por ambos lados se supone salida de la red.

En la página se separan las entradas y las salidas con el fin de poder definir las con mayor facilidad para el usuario final.



Figura 14. Condiciones de contorno de la entrada a la red.

Entre las condiciones de contorno de entrada se puede elegir entre:

- Entrada de caudal en función del tiempo  $Q(t)$ .
- Calado en función del tiempo  $h(t)$ .
- Calado absoluto, desde la cota cero, en función del tiempo  $h+z(t)$ .

En el ejemplo considerado en todas las entradas se tendrá un caudal constante de  $10\text{m}^3/\text{s}$  hasta estabilizarse y posteriormente se introducirá una perturbación tras la cual volverá al caudal anterior. Esto se ilustra en la figura 14.

Navigli permite añadir tantos puntos como se desee que serán mostrados en la gráfica adyacente de manera inmediata [Anexo3, Condiciones de contorno, Javascript línea 207].

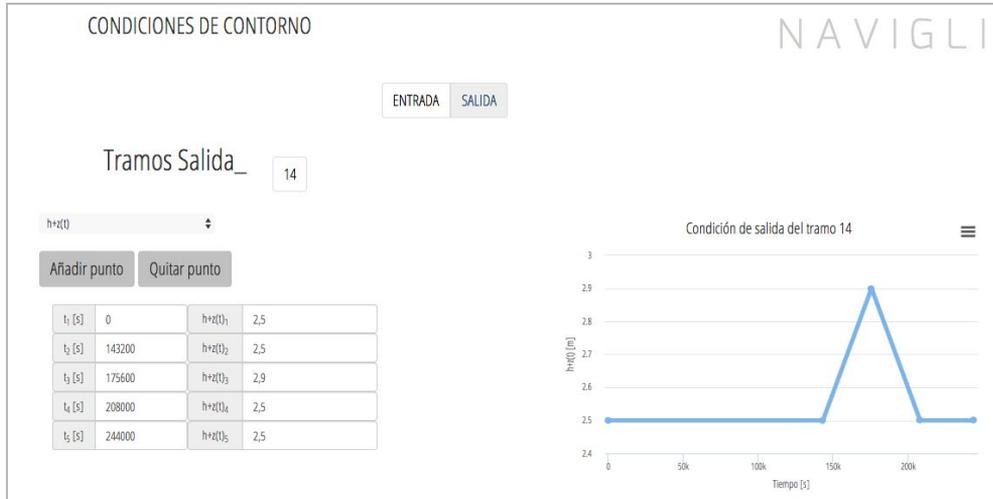


Figura 15. Condiciones de contorno de salida de la red.

En la figura 15 se observan las condiciones de contorno de salida, en concreto de la salida del tramo 14.

Entre estas condiciones de salida se pueden escoger entre:

- Calado en función del tiempo  $h+z(t)$ .
- Caudal en función del calado absoluto  $Q(h+z)$ .
- Salida libre.
- Limitado mediante un número de Froude.

## 5.6. Parámetros

Como último paso de la parte de diseño de la simulación se encuentra la parte de parámetros globales aplicables a toda la red.

PARÁMETROS DE SIMULACIÓN

NAVIGLI

Tiempo inicial de la simulación 0

Tiempo final de la simulación 244000

CFL 0,9

Inter. Data 2440

Simular →

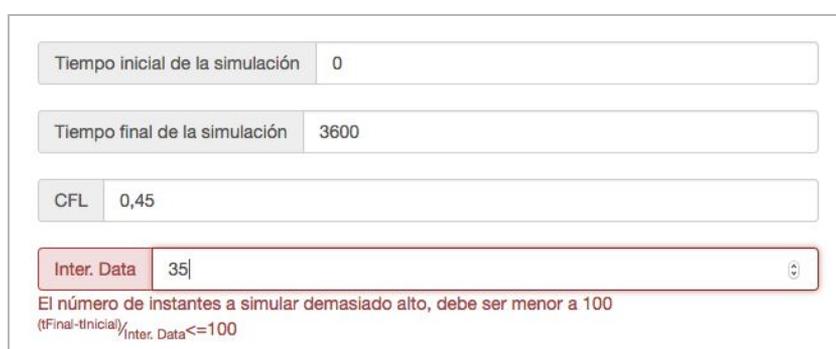
Figura 16. Parámetros de la simulación.

En esta parte se definirán los tiempos iniciales y finales de la simulación. También se definirá el CFL, explicado en el capítulo 2.3.

Por último se encuentra el dato *Inter. Data* es el tamaño del intervalo de tiempo que transcurre entre cada dato [Anexo3, Condiciones de contorno, HTML línea 391].

En el caso que tratamos a lo largo de este trabajo se han fijado los datos a los que se muestran en la figura 16.

El número de archivos a generar está limitado a 100 por tramo para no sobrecargar el servidor en el que se aloja la herramienta, en caso de que el usuario introduzca datos para que genere más no se le permitirá, el botón *Siguiente* estará deshabilitado y se le mostrará un aviso [Anexo3, Condiciones de contorno, Javascript línea 6] tal y como se puede apreciar en el ejemplo de la figura 17.



Tiempo inicial de la simulación	0
Tiempo final de la simulación	3600
CFL	0,45
Inter. Data	35

El número de instantes a simular demasiado alto, debe ser menor a 100  
 $(\text{Final-Tiempo Inicial}) / \text{Inter. Data} \leq 100$

Figura 17. Ejemplo mensaje denegando la generación de más de 100 instantes de simulación por tramo.

Existe también un botón en la esquina superior derecha con información adicional sobre estos parámetros disponible para el usuario.

Mediante estos pasos la herramienta genera unos archivos de texto plano con extensión *.input* los cuales son leídos por la parte del modelo de simulación.

Los archivos que se generan son los siguientes:

- *geometryXYZ.input*: en este archivo se recogen las secciones y los puntos que definen las mismas, así como el número de celdas de cálculo de cada tramo, el calado mínimo y el coeficiente de Manning.
- *junctions.input*: muestra las uniones que tiene el sistema, cuántos canales une y qué canales son. Este archivo es calculado automáticamente por el programa comparando coordenadas, como se ha comentado anteriormente.
- *initial.input*: recoge las condiciones iniciales del sistema, es decir, si los canales están húmedos y con qué calado en este caso o bien si están secos. Se definen por tramo, de esta manera unos tramos se pueden suponer secos y el resto no.
- *inlet.input*: con el dato del tramo o los tramos que son entrada a la red de canales se definen unas condiciones de contorno, dando el número de puntos que el usuario desee.

- *outlet.input*: recoge las condiciones de contorno en las salidas de la red pero a diferencia del caso anterior no se precisa de información adicional para saber qué tramos son salida de la red. *Navigli* lo calcula automáticamente.
- *parameters.input*: define los parámetros de simulación como son el tiempo inicial y final, CFL y el intervalo entre instantes de cálculo.

Estos archivos deben tener la estructura exacta necesaria para que la herramienta de cálculo sea capaz de leerlos, interpretarlos y devolver los resultados oportunos.

## 6. Resultados

La última parte, en la cual se muestran los resultados para su análisis, hay tres gráficas en una misma página, que ofrece al usuario información sobre cualquier punto de la red. En este capítulo se va a explicar cómo utilizar esta página y cómo entender mejor los resultados.

### 6.1. Gráfica de resultados en planta

En esta primera gráfica podemos observar el dibujo de nuestra red en planta y mediante un gradiente de color, podemos visualizar la distribución espacial de variables. La gráfica de planta permite ver cuatro variables, el calado, el caudal, la velocidad o el número de Froude del flujo [Anexo3, Gráficas resultado, HTML línea 209].

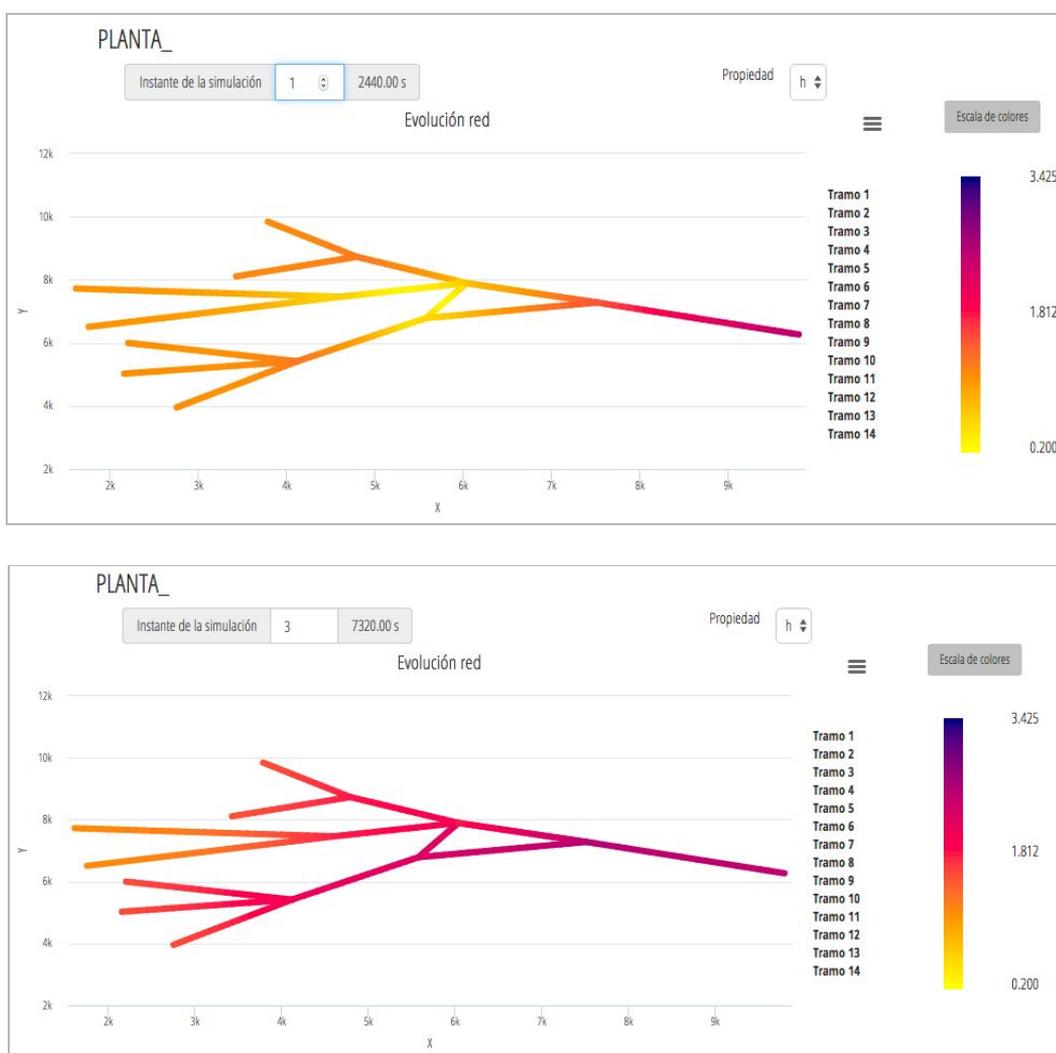


Figura 18. Evolución gráfica gradiente en planta.

En esta gráfica podremos introducir el instante de tiempo en el que queremos visualizar los datos. Dispondremos de tantos instantes de tiempo como haya desde el inicio hasta el fin de la simulación con el intervalo de tiempo introducido en los parámetros.

En el caso del número de Froude la escala de color del gradiente será diferente [Anexo3, Gráficas resultado, Javascript encargado de la interacción con la página, línea 81] ya que se centra la escala en 1, punto en el cual es crítico y por encima es supercrítico y por debajo subcrítico. Se utilizará otra escala de color para hacerlo más visual.

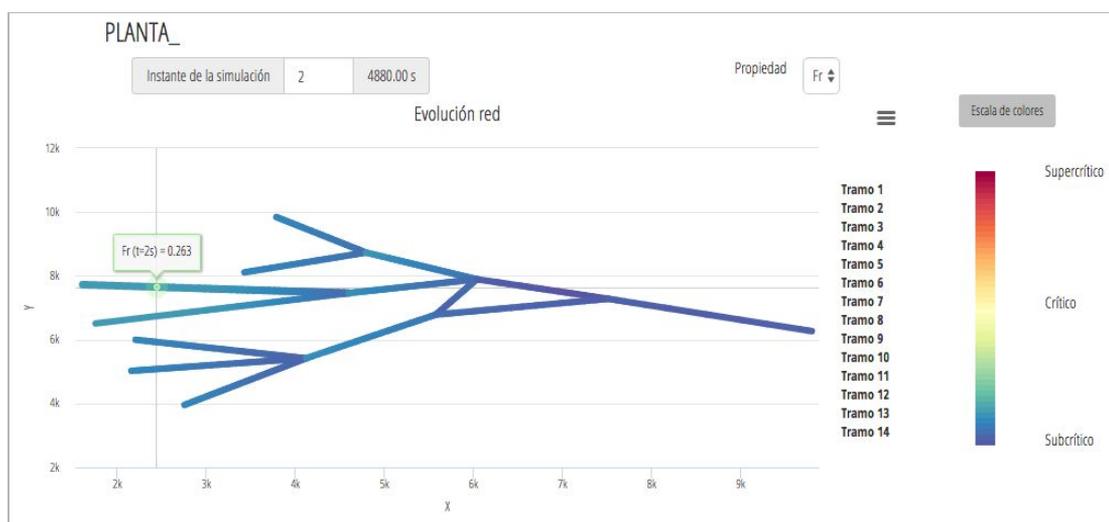


Figura 19. Gráfica en planta observando el número de Froude.

Como en las demás gráficas de esta herramienta, es posible obtener el dato concreto poniendo el puntero sobre el punto que deseemos y podemos hacer zoom sobre la gráfica en el eje X.

El usuario es capaz de dejar de ver algún tramo pulsando en su número en el lateral y de esa manera ver con más claridad el resto de tramos.

Es posible fijar los valores máximo y mínimo de la escala de colores por si se desea ver algo en detalle o hacer alguna comprobación. Para ello deberemos pulsar sobre el botón *Escala de* [Anexo3, Gráficas resultado, HTML línea 226], como se indica en el cuadro de diálogo si los valores están fijados a cero se aplicarán los máximos y mínimos automáticos que se calculan para cada caso concreto como se hace por defecto al ver los resultados. Esto se ilustra en la figura 20.

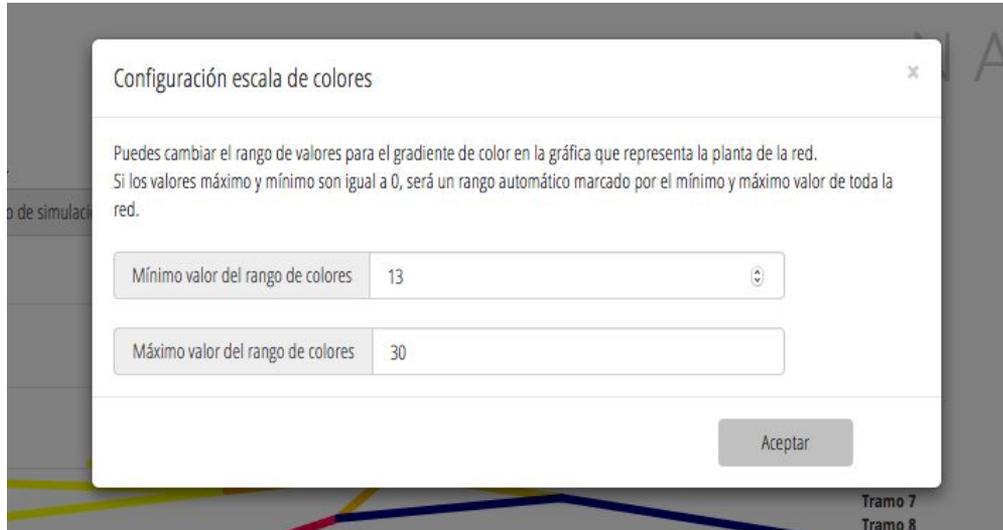


Figura 20. Cuadro de diálogo para modificar el rango de color

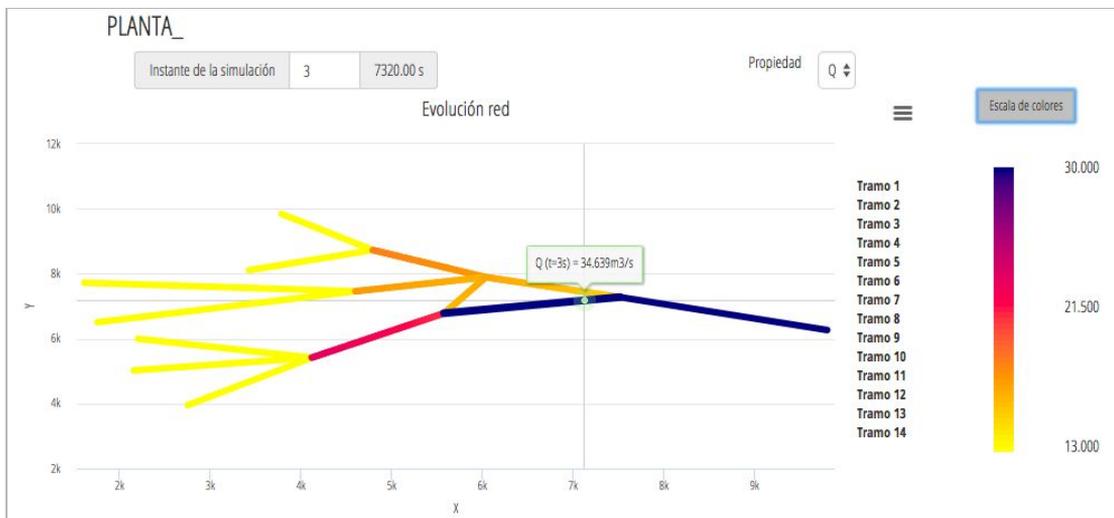


Figura 21. Gráfica de la red con la escala introducida manualmente.

## 6.2. Gráfica a lo largo de la longitud del tramo

En la segunda gráfica se muestra cómo evoluciona una variable a lo largo de toda la longitud de un canal concreto en un tiempo específico. Ambas variables son introducidas por el usuario.

Las propiedades que se pueden observar son:

- $h$ : Calado del flujo medido en metros desde el fondo del canal hasta la superficie libre.
- $Q$ : Caudal del tramo a lo largo de la longitud del canal observado. Medido en  $\text{m}^3/\text{s}$ .
- $u$ : Velocidad, en  $\text{m}/\text{s}$ , del flujo en cada punto a estudiar.
- $A$ : Área mojada del flujo en cada punto. Resultado en  $\text{m}^2$ .
- $z+h$ : A diferencia de  $h$  en esta variable se suma la cota que tiene el fondo del canal y por tanto será mostrado de una manera distinta en la gráfica ya que se mostrará la película

de agua sobre el fondo del canal [Anexo3, Gráficas resultado, Javascript encargado de la interacción con la página, línea 395]. Las unidades serán las mismas que en el caso de  $h$ .

- $Fr$ : Número de Froude, este valor da información sobre el tipo de flujo que se da en cada momento e instante. Es un número adimensional.
- $B$ : Corresponde al ancho del canal, en casos de sección rectangular o cuadrada será un valor constante pero en el resto de los casos será variable. Su resultado se mostrará en metros.

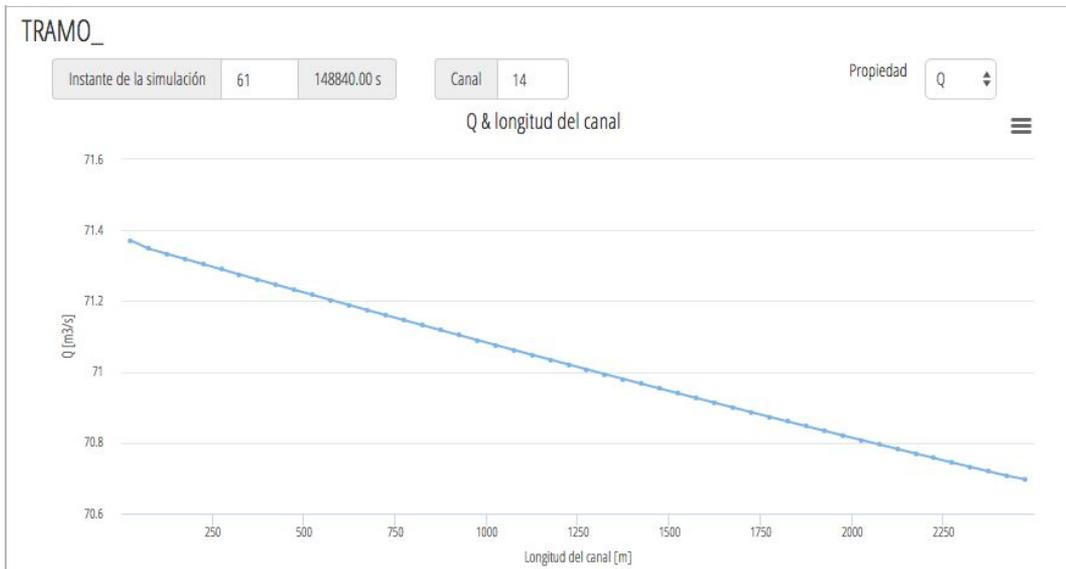
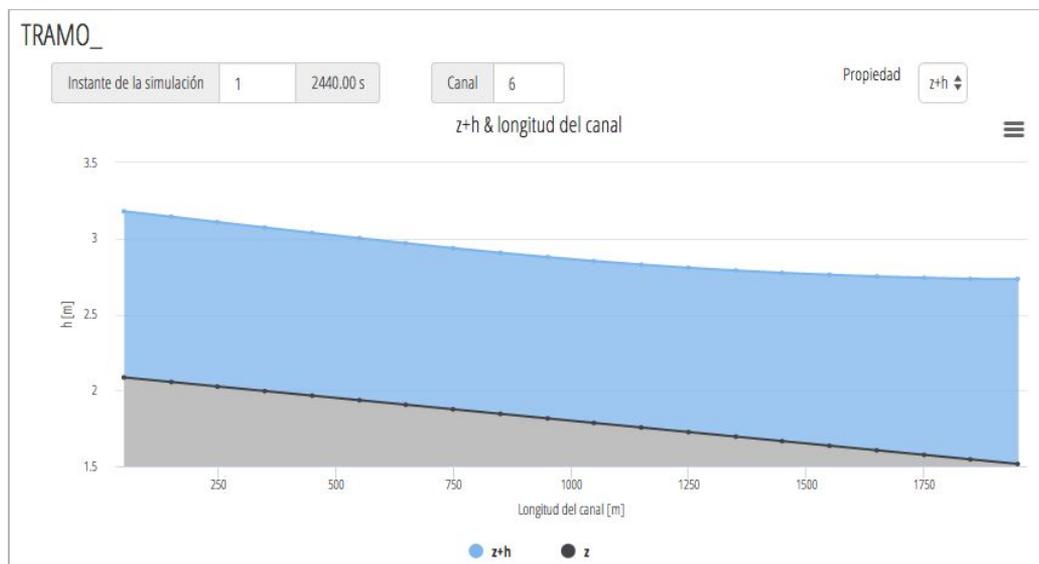


Figura 22. Gráfica a lo largo del tramo.



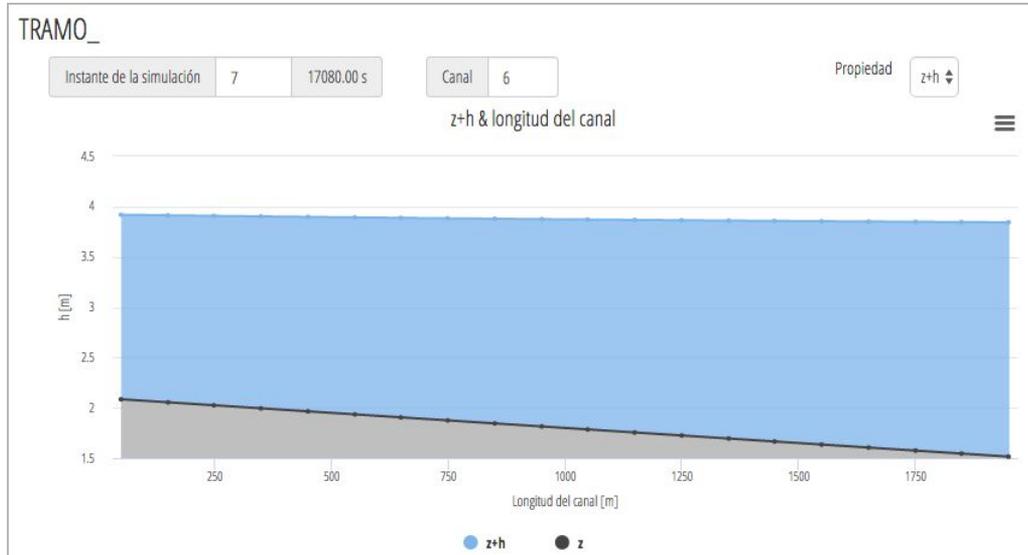


Figura 23. Evolución variable  $z+h$  mostrada en la gráfica a lo largo del tramo.

La gráfica muestra con mucho detalle las variaciones de los datos a lo largo de toda la longitud del tramo, para verlos con más detalle en cualquiera de sus variables es posible hacer zoom y ver las pequeñas variaciones del sistema.

La figura 22 muestra la evolución de  $Q$  a lo largo de la longitud del canal para  $t=148840s$  y la figura 23 muestra la evolución de  $z+h$ , mostrando el fondo del canal, a lo largo del canal para  $t=2440s$  y  $t=17080s$ .

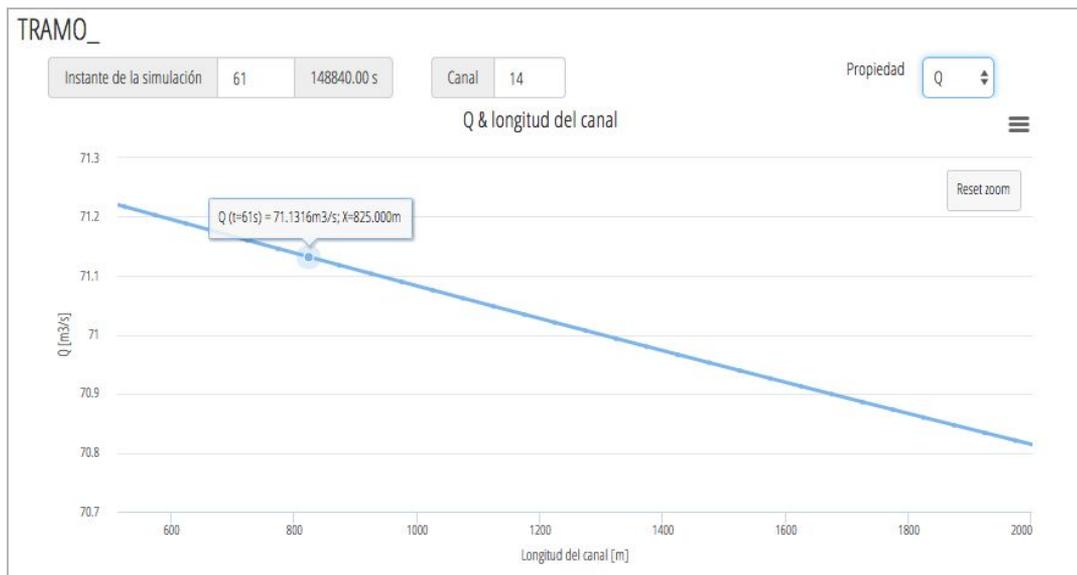


Figura 24. Gráfica (Figura 22) de tramo con el zoom en un rango concreto.

### 6.3. Gráfica de una celda a lo largo de todo el tiempo de la simulación

Se debe escoger un canal y un número de celda concretos [Anexo3, Gráficas resultado, HTML línea 286], mostrándose la distancia de esta celda al inicio del canal, como se muestra en la

figura 25. El número de celdas en cada canal se ha definido en la página de diseño de la topología de cada tramo y puede variar según el tramo.

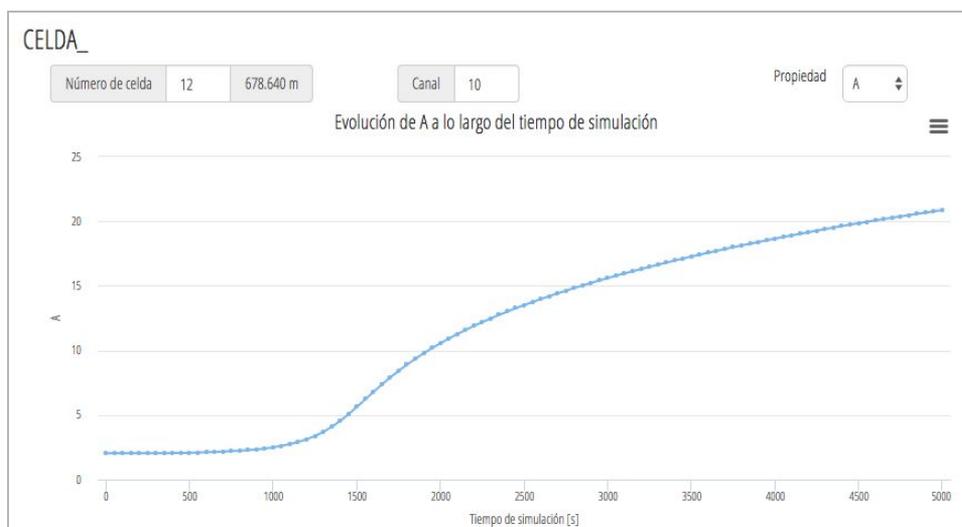


Figura 25. Gráfica de celda a lo largo del tiempo de la simulación.

Todas las gráficas mostradas en la herramienta permiten la descarga de la imagen en cualquier instante y más de un formato para facilitar al usuario final introducirlas en informes o documentos.

## 6.4. Tique y edición de casos

Al final de la página de resultados existe un botón *Editar este caso* [Anexo3, Gráficas resultado, HTML línea 354] que como indica permite que el usuario no tenga que diseñar de nuevo toda la red para hacer un cambio en la simulación. Esto se consigue mediante el nombre del tique que se compone de dos partes, un identificador único y el nombre que el usuario haya decidido ponerle en los primeros paso del diseño.

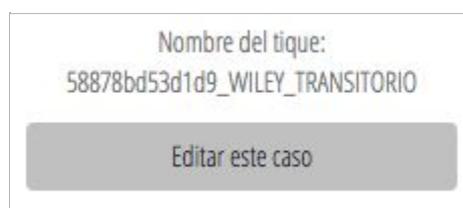


Figura 26. Pie de página de resultados donde se puede comenzar la edición.

El caso creado tiene el nombre que aparece en la figura 26 (58878bd53d1d9\_WILEY\_TRANSITORIO), pudiéndose consultar en cualquier momento accediendo a [canalflowmodel.net/navigli](http://canalflowmodel.net/navigli) de la forma que se explica posteriormente en el capítulo 7.2.

## 6.5. Validación de resultados en *Navigli*

Los resultados se han verificado, mediante las gráficas de las figuras 29, 30, 31 y 32, satisfactoriamente con los resultados presentados en las publicaciones relacionadas<sup>[4][5]</sup>, demostrando, de esta forma, el correcto funcionamiento de la herramienta Navigli.

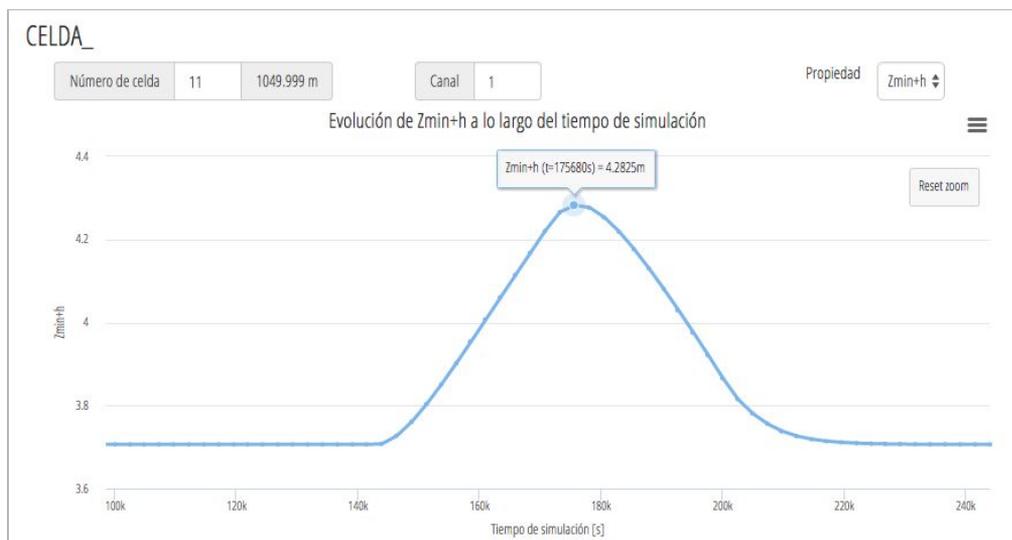


Figura 27. Aumento del calado en la celda 11 del primer canal de 0,6 m aproximadamente.

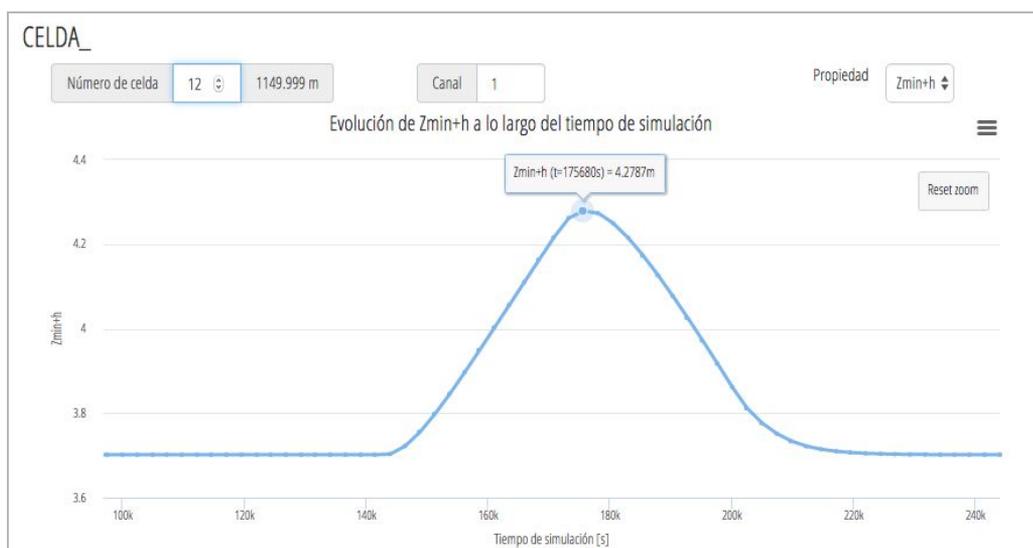


Figura 28. Aumento del calado en la celda 12 del primer canal de 0,6 m aproximadamente.

Las figuras 27 y 28 representan el aumento del calado en el mismo orden de magnitud que el aumento representado en los artículos ya mencionados.

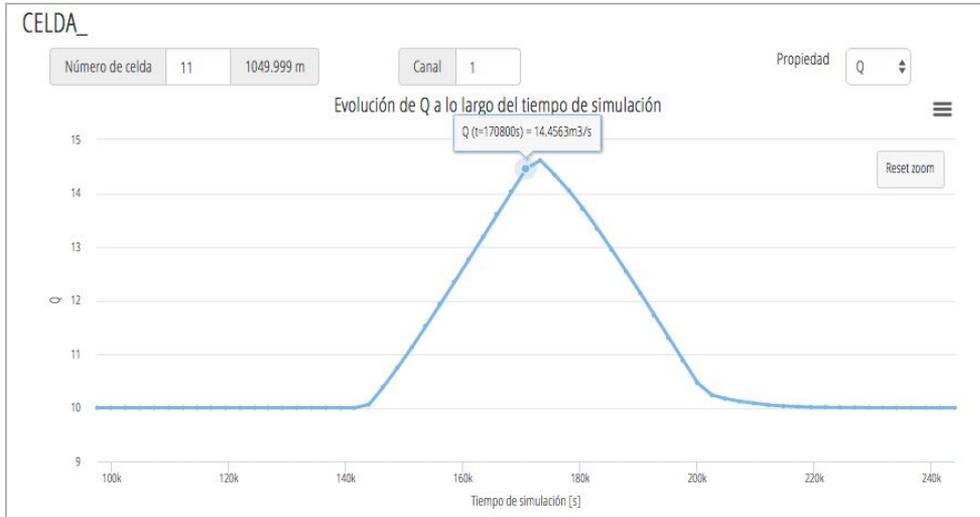


Figura 29. Caudal de la celda 11 del primer canal.

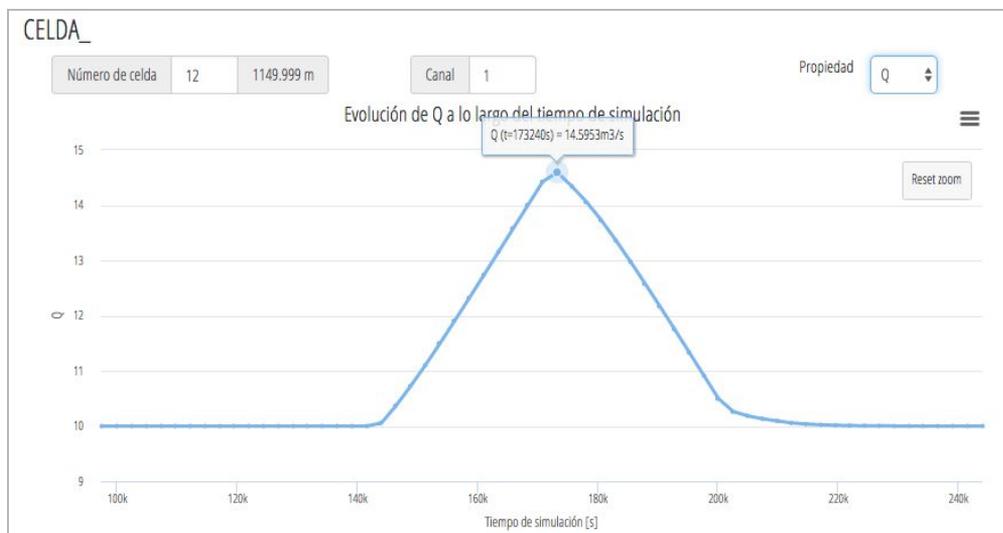


Figura 30. Caudal de la celda 12 del primer canal.

En las figuras 29 y 30 se observa el caudal en las celdas marcadas, que son las estudiadas y coinciden con los resultados de los artículos [\[4\]\[5\]](#) mencionados.

Estos resultados parten de  $t=100.000s$  ya que se le ha dado a la simulación ese tiempo para que parta de una situación estacionaria con los canales estabilizados.

## 7. Atajos de Navigli

En este capítulo se mostrarán una serie de caminos alternativos dentro de la herramienta con el fin de facilitar su uso al usuario final y poder así evitar al usuario la labor de definir casos muy similares más de una vez.

### 7.1. Edición de casos

Una vez el usuario ha pulsado el botón *Editar este caso*, la web le devuelve a la primera página del diseño, en la que se define el número de tramos y se le da nombre al caso. Existen dos posibilidades, que edite el mismo caso o que cree uno distinto pero basado en el caso anterior.

Si el usuario decide editar el caso creado anteriormente debe dejar en la caja de texto de *Pon nombre a tu simulación* el mismo nombre que tenía el anterior tique, en este caso la caja de texto se pondrá de color amarillo para llamar la atención y un mensaje advertirá de que el caso anterior será modificado, como se muestra en la figura 31 [Anexo3, Elección de tramos, Javascript línea 15]. Si el usuario hace cualquier cambio en el nombre, automáticamente desaparecerá esta advertencia, el caso podrá ser guardado con un nombre de tique distinto y podrá conservar los dos.

En el caso en el que quiera crear un caso distinto pero basado en el anterior solamente deberá cambiar el nombre y seguir el curso normal de diseño de una red de canales, los datos del caso anterior rellenarán automáticamente todas las cajas de texto del caso permitiendo modificar cualquier parámetro.

Figura 31. Edición de un caso con el mismo nombre que el original.

Si el usuario decide simular un nuevo caso con un número menor de tramos, por ejemplo, se eliminarán los tramos desde el último al primero.

## 7.2. Visualización rápida de resultados

Debido a que todos los casos son guardados en un servidor, es posible consultar sus resultados sin necesidad de volver a simular el caso completo. Esto es posible gracias a un botón que existe en la página de *Número de tramos*, la primera página tras la página de inicio, en el que se muestra el texto *¿Tienes ya un caso simulado?*. Cuando el usuario presiona ese botón le aparece una ventana superpuesta en la que le explica el proceso para consultar los resultados.

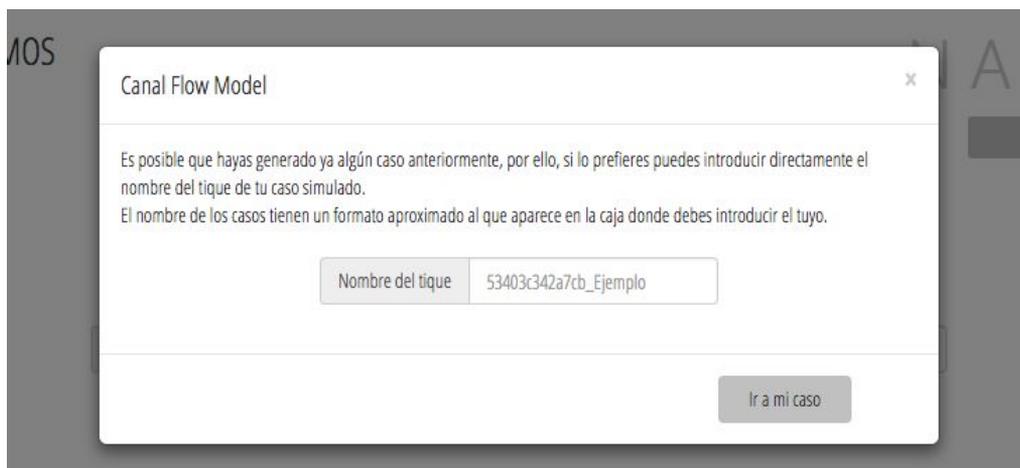


Figura 32. Ventana para la vista rápida de casos generados.

En el momento en el que se presione el botón *Ir a mi caso* [Anexo3, Elección de tramos, HTML línea 68], se recargará la página de resultados mostrando las gráficas correspondientes al caso que se deseaba. Esto se muestra en la figura 32.

## 8. Conclusiones

El objetivo del Trabajo de Fin de Grado era principalmente el desarrollo una herramienta web con el fin de simplificar y acelerar el complejo proceso de diseño de una red de canales. Basado en la herramienta para el estudio de un solo tramo llamado CanalFlowModel desarrollada en el Área de Mecánica de Fluidos trata ampliar su campo de aplicación. Pudiéndose también trabajar conjuntamente con las dos herramientas ya que la herramienta permite mayor número de variables y propiedades.

El proceso de diseño está simplificado al máximo en cuanto a la demanda de datos por parte del programa se refiere de manera que carga con un mayor grado de complejidad de cálculos al programa.

Otro asunto a tener en cuenta es el conjunto de lenguajes de programación utilizados en el desarrollo de esta herramienta ya que utiliza un equilibrio de lenguajes del lado del servidor y del cliente de manera que optimiza el reparto de carga de trabajo entre ambos lados. Se ha realizado por parte del autor del trabajo un aprendizaje desde cero de todos y cada uno de los lenguajes de programación utilizados hasta el dominio de todos ellos.

El balance del aprendizaje obtenido mediante la realización de este trabajo es muy positivo tanto en lo relativo a la mecánica de fluidos como a las herramientas informáticas que se han utilizado para su elaboración.

En el futuro se deberán afrontar mejoras en esta herramienta y en su modelo de simulación para su constante evolución con el fin de poder hacer un programa que permita generar casos más personalizables tanto en tipos de secciones transversales como en parámetros y de esta manera enriquecer el desarrollo tecnológico en cuanto a la mecánica de fluidos se refiere.

# Bibliografía

- [1] Área de Mecánica de Fluidos de la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza, *Herramienta [CanalFlowModel](#)*, 2014
- [2] Hydrologic Engineering Center del US Army Corps of Engineers, <http://www.hec.usace.army.mil/software/hecras>, 1995
- [3] DHI (Institute for Water and Environment), <https://www.mikepoweredbydhi.com/products/mike-11>, 2003
- [4] Song, Lixiang, entre otros, *An Efficient One-Dimensional Hydrodynamic Model for Complex Tidal River Networks*, International Journal for Numerical Methods in Fluids, 2014
- [5] Islam, Adlul, entre otros, *Comparison of Gradually Varied Flow Computation Algorithms for Open-Channel Network*, 2005
- [6] Fernández Pato, Javier y García-Navarro Pilar, *Modelo de redes de Flujo con Transición dinámica de régimen lámina libre/presión*, 2012
- [7] García-Navarro, Pilar and A. Priestley, *The application of an implicit TVD method for water flow modelling in channels and pipes*, 1993

# Anexos

## Anexo 1. Validación caso estacionario

### Modificación. Validación estacionaria

En este caso se simula la misma red a nivel geométrico pero con las condiciones de contorno cambiadas (partiendo de seco), como se muestra en la figura I, para todos los tramos se introduce lo mismo, un caudal constante  $Q=10m^3/s$ . La salida, por el tramo 14, es libre.



Figura I. Condición de contorno en la entrada 1. Igual en las demás entradas.

Con esta simulación se trata de mostrar la continuidad existente en la red, es decir, el caudal en la salida debe ser la suma de todas las entradas. Esto se muestra en la figura II que muestra el caudal de la última celda del canal de salida a lo largo de todo el tiempo de la simulación.

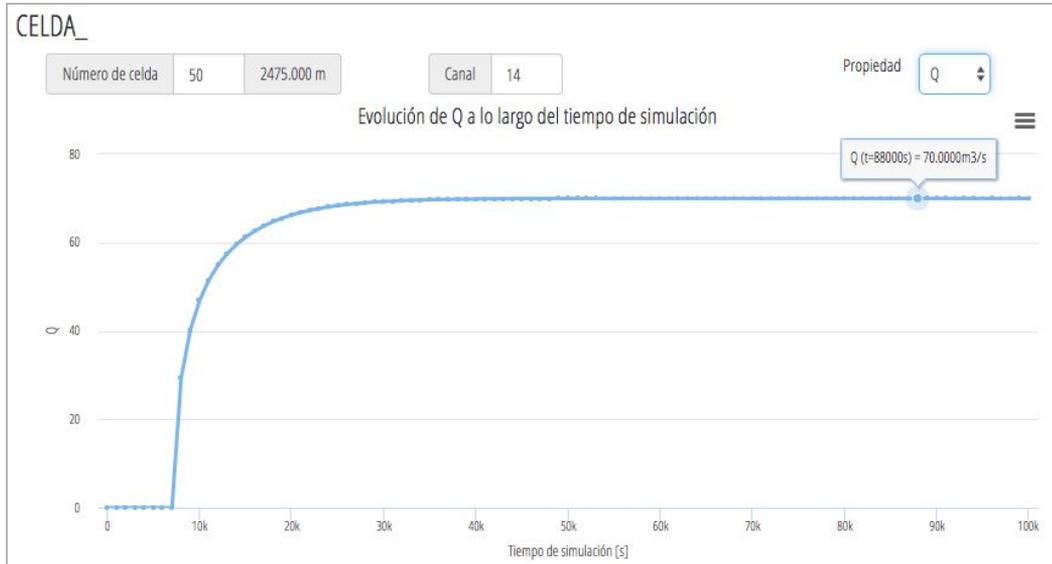


Figura II. Evolución de la última celda del canal 14.

Como tratábamos de verificar, se puede observar en la figura II que para  $t=8800s$  se puede decir que se ha estabilizado totalmente la red.

Se adjunta en la figura III la visión general de la red en ese instante.

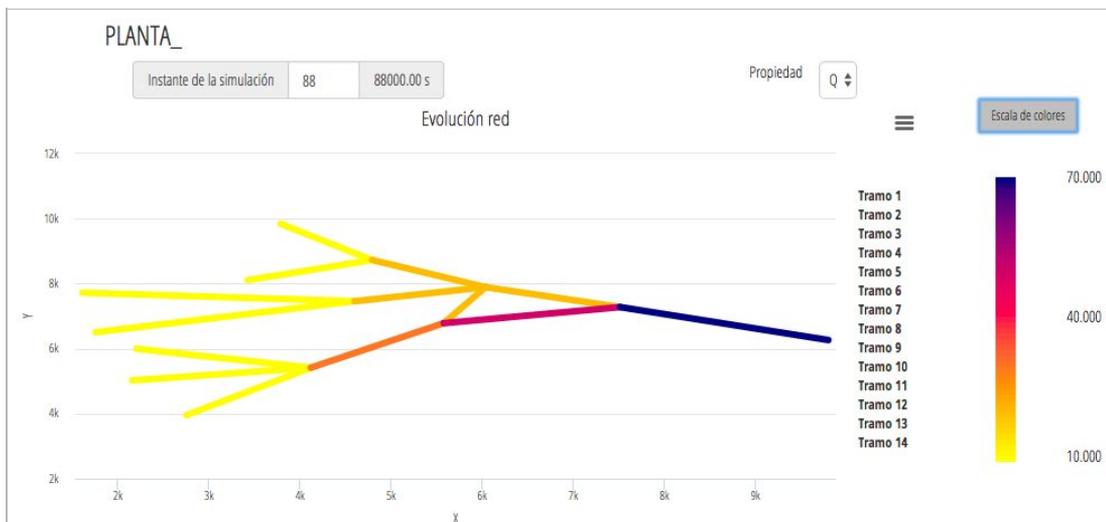


Figura III. Visión general de la red estabilizada.

## Anexo 2. Manual de usuario de Navigli

El manual de usuario es un apoyo rápido para la utilización de esta herramienta y se encuentra disponible en la página de inicio de Navigli ([http://canalflowmodel.net/navigli/Manual\\_NAVIGLI.pdf](http://canalflowmodel.net/navigli/Manual_NAVIGLI.pdf)) para que el usuario pueda disponer de él.

# NAVIGLI

## SIMULACIÓN DE REDES HIDRÁULICAS Manual de uso rápido

Juan C. de Val Cremades

## Instrucciones de uso

En este manual se explicará el funcionamiento de Navigli así como las posibilidades que tiene.

Los requisitos para su utilización son los siguientes:

### Acceso a internet

Desde cualquier dispositivo conectado a internet se puede simular una red de canales. Los navegadores soportados son los siguientes:

- Internet explorer v6.0+
- Firefox v2.0+
- Chrome v1.0+
- Safari v4.0+
- Opera v9.0+
- Android browser v 2.0+

### Geometría

El usuario deberá disponer de los datos geométricos y topológicos tanto de las secciones como de los tramos que conforman la red.

### Condiciones de contorno Entrada y Salida

La red debe tener definidas tanto las condiciones de contorno de entrada como las de salida para poder simular la red diseñada.

### Parámetros

Parámetros como el coef. de Manning, el CFL, el tiempo de simulación, el calado mínimo e inicial y número de celdas de cálculo por tramo con necesarios a la hora de diseñar una simulación.

# Datos para leer este manual

## Datos relevantes

- Coeficiente de Manning: coeficiente de rugosidad de las paredes del canal.
- Calado inicial: profundidad de agua inicial en el canal con el que se comienza la simulación.
- Calado mínimo: profundidad el agua límite en la que por debajo de ese valor, la velocidad del agua se supone cero.
- CFL: Número de Courant-Friedrich-Levy.
- Número de celdas: cantidad de partes en las que se divide cada tramo para su cálculo.
- Q: caudal.
- h: calado desde el fondo del canal.
- $Z_{\min} + h$ : calado desde la cota cero.
- b: ancho de la superficie libre del agua.
- A: área mojada de la sección transversal.
- Fr: número de Froude.
- U: velocidad del agua.

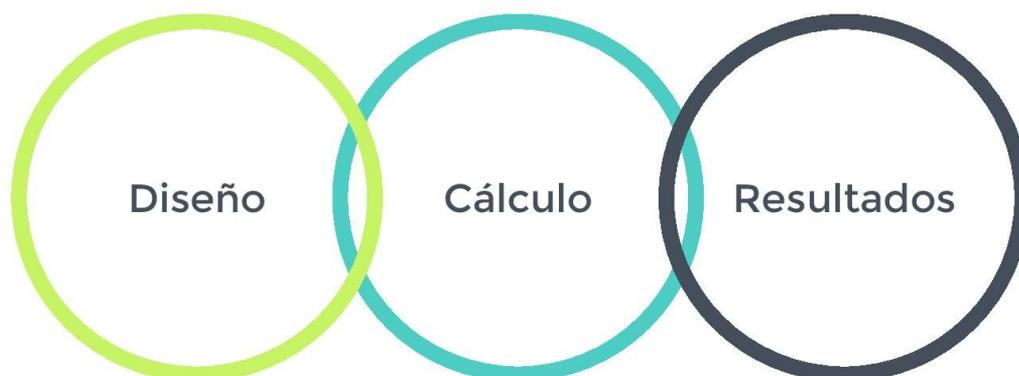


## Inicio

Para empezar una nueva red sólo hay que pulsar *Comenzar* en la página de inicio

## Pasos en Navigli

---



# Diseño

## de la red de canales

---

Primeros pasos

# 1.

## Elección de tramos y nombre

### Tramos y nombre

- Introduce el número de tramos de tu red

Número de tramos de la red 7

- Pon nombre a tu simulación

Pon nombre a tu simulación Prueba1

- Visualiza casos ya generados

¿Tienes ya un caso simulado?

# 2.

## Forma de las secciones

### Forma de las secciones

- Rectangular/trapezoidal
- Semicircular
- Triangular

Introduce las medidas de la sección perpendicular al flujo de cada tramo así como su coeficiente de Manning.

La forma se visualizará en una gráfica en cuanto rellenes los datos. Aparecerá el botón *Siguiente* en el último tramo. Esto se repetirá en toda la herramienta

¿Qué forma tiene la sección del tramo 1?

Rectangular/trapezoidal

Longitud del borde superior metros

Altura metros

Longitud de la base metros

Continúe definiendo el tramo 2

Número de Manning del tramo 1 0,01

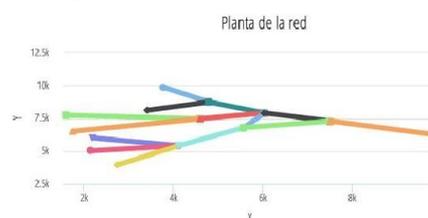
# 3.

## Topología

### Topología

- Coordenadas de entrada y salida de cada tramo, tienes información sobre estos puntos pulsando el botón ⓘ
- Señala los tramos que son entrada.
  - El tramo 1 es entrada de la red.
- Si lo deseas, tienes una configuración avanzada ⓘ ahí encontrarás calado inicial y mínimo así como el número de celdas de cálculo.

Para que exista una unión, las coordenadas X e Y de dos tramos deben coincidir.

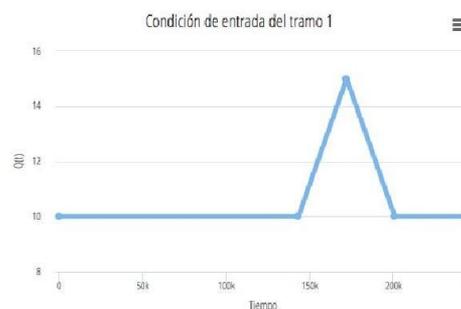


# 4.

## Condiciones de contorno

### Condiciones de contorno

- Escoge entre entrada y salida
- Selecciona el tipo de condición
  - Entrada
    - $Q(t)$
    - $h(t)$
    - $h+z(t)$
  - Salida
    - $h(t)$
    - $Q(h+z)$
    - Salida libre
    - Número de Froude
- Introduce los datos que necesites para tu simulación



# 5.

## Parámetros

### Parámetros

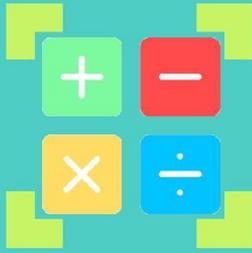
Debes introducir los siguientes parámetros

- Tiempo inicial de la simulación
- Tiempo final de la simulación
- CFL (debe ser menor o igual a 1)
- Intervalo de tiempo entre datos. No podrán generarse más de 100 datos.

Tiempo inicial de la simulación	0
Tiempo final de la simulación	360000
CFL	0,9
Inter. Data	3600

Si tienes alguna duda sobre estos parámetros pulsa en  para más información.

Si has terminado, pulsa *Simular* para ver los resultados



# Cálculo

En este paso Navigli está calculando la simulación introducida

# Resultados

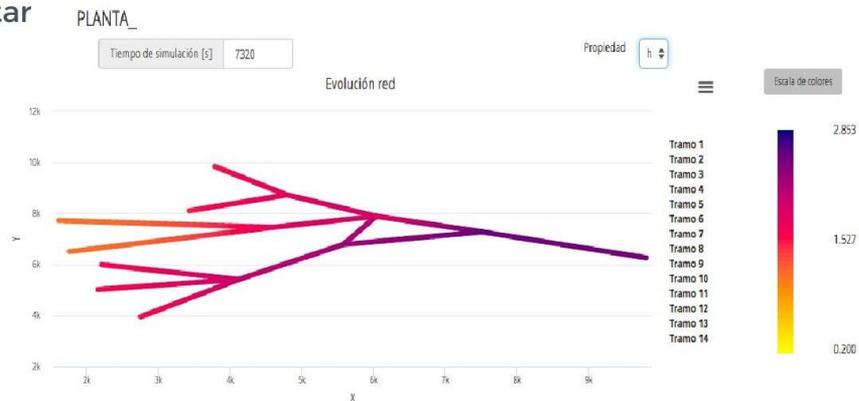
de tu simulación

---

# 1. Planta

## Gráfica planta

- Escoge el tiempo y la propiedad que deseas ver (h, Q, U o Fr). La escala de color es automática.
- Puedes fijar la escala de color a tu elección. Escala de colores
- Ponte sobre el punto que te interese para ver su valor y haz clic para hacer zum. Pulsa el número del tramo que deseas ocultar

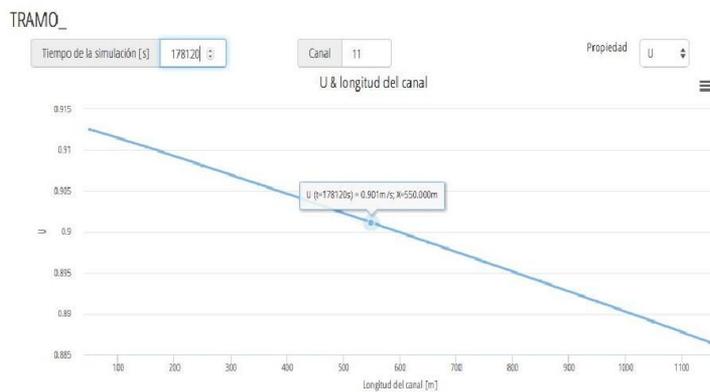


# 2.

## Tramo

### Gráfica tramo

- Podrás observar cómo evoluciona una propiedad a lo largo de toda la longitud del tramo seleccionado en el tiempo que selecciones.
- Propiedades que puedes ver:  $h$ ,  $Q$ ,  $U$ ,  $A$ ,  $b$ ,  $Z_{\min} + h$  o  $Fr$ .
- Ponte sobre el punto que te interese para ver su valor y haz clic para hacer zoom.



# 3.

## Celda

### Gráfica celda

- Podrás observar cómo evoluciona una propiedad a lo largo de todo el tiempo de la simulación en la celda y tramo que selecciones.
- Propiedades que puedes ver:  $h$ ,  $Q$ ,  $U$ ,  $A$ ,  $b$ ,  $Z_{\min}+h$  o  $Fr$ .
- Ponte sobre el punto que te interese para ver su valor y haz clic para hacer zum.



## Editar caso

- Para editar una simulación y guardarlo en el mismo tique o en otro distinto sólo tienes que pulsar el botón de la página de resultados:

Nombre del tique:

58878bd57d1d9\_NombreCaso

Editar este caso

Luego sólo tienes que seguir los mismos paso anteriormente descritos.

# GRACIAS

**¡Gracias por utilizar Navigli!**

Área de Mecánica de Fluidos  
Escuela de Ingeniería y Arquitectura  
Universidad de Zaragoza



### Anexo 3. Código fuente del programa

A continuación se adjunta el código fuente de la página en el orden de visualización de la herramienta. Todo el código mostrado ha sido desarrollado por el autor de este trabajo.

## HTML

```

1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7
8     <link rel="apple-touch-icon" sizes="57x57"
... href="css/img/apple-icon-57x57.png">
9     <link rel="apple-touch-icon" sizes="60x60"
... href="css/img/apple-icon-60x60.png">
10    <link rel="apple-touch-icon" sizes="72x72"
... href="css/img/apple-icon-72x72.png">
11    <link rel="apple-touch-icon" sizes="76x76"
... href="css/img/apple-icon-76x76.png">
12    <link rel="apple-touch-icon" sizes="114x114"
... href="css/img/apple-icon-114x114.png">
13    <link rel="apple-touch-icon" sizes="120x120"
... href="css/img/apple-icon-120x120.png">
14    <link rel="apple-touch-icon" sizes="144x144"
... href="css/img/apple-icon-144x144.png">
15    <link rel="apple-touch-icon" sizes="152x152"
... href="css/img/apple-icon-152x152.png">
16    <link rel="apple-touch-icon" sizes="180x180"
... href="css/img/apple-icon-180x180.png">
17    <link rel="icon" type="image/png" sizes="192x192"
... href="css/img/android-icon-192x192.png">
18    <link rel="icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png">
19    <link rel="icon" type="image/png" sizes="96x96"
... href="css/img/favicon-96x96.png">
20    <link rel="icon" type="image/png" sizes="16x16"
... href="css/img/favicon-16x16.png">
21    <link rel="stylesheet" href="css/bootstrap.min.css">
22    <link href="fonts/googleFonts.woff2" rel="stylesheet">
23    <link rel="stylesheet" href="css/main.css">
24    <script src="js/jquery.min.js"></script>
25    <script src="js/bootstrap.min.js"></script>
26
27    <meta charset="UTF-8">
28
29
30    <title>Canal Network Flow</title>
31 </head>
32
33 <body>
34     <div>
35         <br>
36     </div>
37     <div class="row">
38         <div class="col-xs-10"></div>
39         <button type="button" class=" btn glyphicon glyphicon-info-sign btn-sm "
```

```

39... data-toggle="modal" data-target="#info" title="Info Canal Flow Model"></button>
40   <a href="en/"></a>
41
42   <div class="modal fade letra" id="info">
43     <div class="modal-dialog" role="document">
44       <div class="modal-content">
45         <div class="modal-header">
46           <button type="button" class="close" data-dismiss="modal"
... aria-label="Close">
47             <span aria-hidden="true">&times;</span>
48           </button>
49           <h4 class="modal-title" id="myModallabel">Canal Network
... Flow</h4>
50         </div>
51         <div class="modal-body">
52           &nbsp; &nbsp; &nbsp; &nbsp; Te presentamos Navigli, una
... herramienta mediante la cual podrás simular el flujo de agua por redes de
... canales. En esta herramienta podrás escoger la sección de los canales, la
... topología de los canales, las condiciones de contorno y más parámetros que se
... utilizan para generar una red de canales a tu medida.
53           </br>
54           Dispones de un rápido manual para aclarar tus dudas
... sobre Navigli.
55           </br>
56           </br>
57           </br>
58           <a class="glyphicon glyphicon-save"
... href="Manual_NAVIGLI.pdf" target="_blank"> Manual de uso</a>
59
60         </div>
61         <div class="modal-footer">
62           <a class="col-xs-1" target="_blank"
... href="https://eina.unizar.es/"></a>
63           <div class="col-xs-9"></div>
64           <button type="button" class="btn btn-secondary col-xs-2"
... data-dismiss="modal">Entendido!</button>
65         </div>
66       </div>
67     </div>
68   </div>
69 </div>
70 </div>
71 <div class="row">
72   <h1 class="col-xs-12
... titulo-home">N&nbsp;&nbsp;&nbsp;A&nbsp;&nbsp;&nbsp;V&nbsp;&nbsp;&nbsp;I&nbsp;&nbsp;&nbsp;G&nbsp;&nbsp;&nbsp;L&
... &nbsp;&nbsp;&nbsp;I</h1>
73 </div>
74 <div class="row">
75   <h3 class="col-xs-12 subtitulo-home">SIMULACIÓN &nbsp;&nbsp;&nbsp;DE
... &nbsp;&nbsp;&nbsp;REDES &nbsp;&nbsp;&nbsp;HIDRÁULICAS</h3>
76 </div>

```

```
77     <form class="col-xs-10" action="tramos.php">
78         <input type="submit" class="btn boton-home boton-home-location"
... value="COMENZAR">
79     </form>
80 </body>
81
82 </html>
```

## Elección de tramos

### HTML

```
1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <meta charset="UTF-8">
8     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
9     <link rel="stylesheet" href="css/bootstrap.min.css">
10    <link href="fonts/googleFonts.woff2" rel="stylesheet">
11    <link rel="stylesheet" href="css/main.css">
12    <script src="js/jquery.min.js"></script>
13    <script src="js/bootstrap.min.js"></script>
14    <script type="text/javascript">
15        $(document).ready(function() {
16            $("input#numTramos").focus();
17
18        });
19    </script>
20    <script src="js/editFunciones.js"></script>
21
22    <title>Tramos</title>
23 </head>
24
25 <body>
26     <?php
27     echo'
28 <div class="row">
29     <div class="col-xs-1"></div>
30     <h2 class="col-xs-4 paso-plantilla">NUMERO DE TRAMOS</h2>
31     <h1 class="col-xs-6
... titulo-plantilla">N&nbsp;A&nbsp;V&nbsp;I&nbsp;G&nbsp;L&nbsp;I</h1>
32     <div class="col-xs-1"></div>
33
34 </div>
35 <div class="row">
36 <div class="col-xs-9"></div>
37 <button type="button" id="casoSimulado" class="col-md-2 btn btn-sm letra"
... data-toggle="modal" data-target="#info" title="Resultados de tu caso">¿Tienes
... ya un caso simulado?</button>
38 <div class="col-xs-1"></div>
39
40 </div>
41 <div class="modal fade letra" id="info">
42     <div class="modal-dialog" role="document">
43         <div class="modal-content">
44             <div class="modal-header">
45                 <button type="button" class="close"
... data-dismiss="modal" aria-label="Close">
46                     <span aria-hidden="true">&times;</span>
47                 </button>
```

```

48         <h4 class="modal-title" id="myModalLabel">Canal Flow
... Model</h4>
49     </div>
50     <div class="modal-body">
51         Es posible que hayas generado ya algún caso
... anteriormente, por ello, si lo prefieres puedes introducir directamente el
... nombre del tique de tu caso simulado.
52         <br>
53         El nombre de los casos tienen un formato aproximado al
... que aparece en la caja donde debes introducir el tuyo.
54         <br>
55         <br>
56         <form action="graficaPlanta.php" method="post"
... enctype="multipart/form-data">
57             <div class="col-xs-3"></div>
58             <div class="input-group col-xs-6 letra
... text-center">
59
60                 <span class="input-group-addon" >Nombre del
... tique</span>
61
62                 <input class="form-control" id="nombreTique"
... type="text" placeholder="53403c342a7cb_Ejemplo" name="carpeta"
... required="required" autocomplete="on" />
63             </div>
64             <br>
65         </div>
66         <div class="modal-footer">
67             <div class="col-xs-9"></div>
68             <input class="btn btn-secondary col-xs-2" type="submit"
... value="Ir a mi caso">
69
70         </form>
71     </div>
72 </div>
73 </div>
74 </div>
75
76 <form action="seccion.php" method="post" enctype="multipart/form-data">
77
78
79
80     <div class="row espacio-arriba">
81         <div class="col-xs-3"></div>
82
83
84         <div class="col-xs-6">
85
86             <br>
87
88             <div class="input-group">
89

```

```

90         <span class="input-group-addon letra">Número de
... tramos de la red</span>
91
92         <input class="form-control" type="number"
... name="numTramos" id="numTramos" required="required" autocomplete="on" value="3"
... min="1" />
93     </div>
94
95     </br>
96     </br>
97     </br>
98     </br>
99     </br>
100    </br>
101    </br>
102    </br>
103    </br>
104    <div id="divNombreCaso">
105    <div class="input-group nombreCaso">
106
107        <span class="input-group-addon letra">Pon
... nombre a tu simulación</span>
108
109        <input class="form-control" type="text"
... id="nombreCaso" name="nombreCaso" required="required" autocomplete="on"
... required="required" value="Prueba1" />
110    </div>
111    </div>
112    </div>
113    <div class="col-xs-3"></div>
114    </div>';
115    if (isset($_POST['edit'])) {
116        echo'
117        <input type="number" id="editTramos" name="editTramos"
... value="' . $_POST['edit'] .'" hidden>
118        <input type="text" id="carpetaEdit" name="carpetaEdit"
... step="any" value="' . $_POST['carpeta'] .'" hidden>
119        <input type="number" id="numTramosEdit" name="numTramosEdit"
... value="' . $_POST['numTramos'] .'" hidden>;
120    }
121    echo'
122        <br/>
123        <input class="btn boton-siguiente boton-letra"
... type="submit" value="Siguiente &rarr;"/>
124    </form>
125    '
126    ?>
127 </body>
128
129 </html>

```

## Javascript

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 $(document).ready(function() {
4     function editTramos() {
5         if ($("#editTramos").length > 0) {
6             $("#numTramos").attr("value", parseInt($("#numTramosEdit").val()));
7
8             $("#nombreCaso").attr("value", $("#carpetaEdit").val());
9             //añadir warning que si pones el mismo caso sobrescribirá
10            //el caso anterior pero si lo cambias, crearás un caso nuevo
11            //realmente si cambias el numero de tramos cambias todas los datos
... de después
12        }
13    }
14
15    function warningNombre() {
16        //Avisos que si cambias el numtramos se eliminarán los últimos
17        if ($("#editTramos").length > 0) {
18            if ($("#nombreCaso").val() == $("#carpetaEdit").val()) {
19                $(".nombreCaso").attr("class", "input-group nombreCaso
... has-warning");
20                $("#divNombreCaso").append($("#<p class=\"text-warning\"
... id=\"warningMessage\">Si mantiene el nombre del caso, éste se modificará. Si
... escribe otro, guardará ambos casos.</p>"));
21            } else {
22                $(".nombreCaso").attr("class", "input-group nombreCaso");
23                $(".text-warning").remove();
24            }
25        }
26    }
27    editTramos();
28    warningNombre();
29    $("#nombreCaso").change(function() {
30        warningNombre();
31    })
32 })
```

## Forma de las secciones

### HTML

```
1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <meta charset="UTF-8">
8     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
9     <link rel="stylesheet" href="css/bootstrap.min.css">
10    <link href="fonts/googleFonts.woff2" rel="stylesheet">
11    <link rel="stylesheet" href="css/main.css">
12    <script src="js/jquery.min.js"></script>
13    <script src="js/bootstrap.min.js"></script>
14    <script src="js/highcharts.js"></script>
15    <script src="js/exporting.js"></script>
16    <script src="js/paginacion.js"></script>
17    <script src="js/refrescarSecciones.js"></script>
18
19    <title>Secciones</title>
20 </head>
21
22 <body>
23     <?php
24
25 $numTramos = $_POST['numTramos'];
26 $nombreCaso = $_POST['nombreCaso'];
27
28 //El primer div es para centrar y colocar en un sitio especifico y el ul es
... para el fondo de los botones de la paginacion. El li es para los botones de
... paginación (pinta todos).
29 echo '
30 <div class="row">
31     <div class="col-xs-1"></div>
32     <h2 class="col-xs-4 paso-plantilla">FORMA DE LA SECCIÓN</h2>
33     <h1 class="col-xs-6
... titulo-plantilla">N&nbsp;A&nbsp;V&nbsp;I&nbsp;G&nbsp;L&nbsp;I</h1>
34     <div class="col-xs-1"></div>
35 </div>
36
37 <div class="row">
38 <div class="col-xs-3 tamaño-letra alinear-derecha">Tramos_</div>
39 <div class="col-xs-7">
40     <ul class="pagination letra mediano">';
41     for ($tram=1;$tram<=$numTramos;$tram++){
42     echo '
43         <li><a id="myLink" href="#"
... onclick="prueba('.$tram.', '.$numTramos.');">'.$tram.'</a></li>
44         ';
45     }
46     echo '
47     </ul>
```

```

48 </div>
49 <div class="col-xs-2"></div>
50 </div>';
51
52     echo '
53     <form class="letra" action="topologia.php" method="post"
... enctype="multipart/form-data">
54         ';
55
56         for ($tram=1; $tram<=$numTramos; $tram++){
57
58             //Esto hace que se muestre la primera vez los elementos del
... tramo 1.
59             if($tram==1){
60                 echo '
61                 <div id="tramo' . $tram . '" style="display:block;">';
62                 }else{
63                 echo '
64                 <div id="tramo' . $tram . '" style="display:none;">';
65                 }
66
67                 $numSec = 1;
68                 for ($sec=1; $sec<=$numSec; $sec++){
69
70                     echo '
71                     <div class="row">
72                     <div class="col-xs-1"></div>
73                     <b class="col-xs-3">¿Qué forma tiene la sección <!--
... '$sec.' --> del tramo '$tram.'?</b>
74                     </div>
75
76                     <div class="row">
77                     <div class="col-xs-1"></div>
78                     <select class="btn selectpicker col-xs-3"
... data-toggle="dropdown" name="forma' . $tram . '-' . $sec . '"
... id="selecttipo' . $tram . '-' . $sec . '"
... onchange="refreshCurrentViewTramo('$tram.', '$sec.');" >
79
80                         <option value="0">Selecciona un tipo de
... sección</option>
81
82                         <option value="1">Semicircular</option>
83                         <option
... value="2">Rectangular/trapezoidal</option>
84                         <option value="3">Triangular</option>
85                         </select>
86
87                         <div class="col-xs-1"></div>
88                         </div>
89                         <div class="col-xs-1"></div>
90
91                         <div class="col-xs-5">
<!--/////////////////SEMICIRCULAR/////////////////-->

```

```

92
93         <div id="semicircular'.'.$tram.'-'. '$sec.'"
... style="display: none;" class="col-xs-11">
94
95             <br/>
96
97             <div class="input-group">
98                 <span class="input-group-addon " >Radio del
... semicírculo</span>
99
100                 <input type="number" step="any"
... name="radio'.'.$tram.'-'. '$sec.'" class=" form-control datosSeccion"
... aria-describedby="radio'.'.$tram.'-'. '$sec.'" placeholder="metros" min="0"
... id="radio'.'.$tram.'-'. '$sec.'">
101
102             </div>';
103             if ($tram != $numTramos) {
104                 echo '
105                 <br/>
106                 <p class="bg-info text-center">Continúe
... definiendo el tramo ' .($tram+1).'</p>';
107             }
108             echo '
109
110         </div>
111
112
113 <!--//////////////////TRAPEZOIDAL//////////////////-->
114
115         <div id="trapezoidal'.'.$tram.'-'. '$sec.'"
... style="display: none;" class="col-xs-11">
116
117             <br/>
118
119             <div class="input-group">
120                 <span class="input-group-addon">Longitud
... del borde superior</span>
121
122                 <input type="number" step="any"
... name="supTrapezio'.'.$tram.'-'. '$sec.'" class=" form-control datosSeccion"
... aria-describedby="supTrapezio'.'.$tram.'-'. '$sec.'" placeholder="metros" min="0"
... id="supTrapezio'.'.$tram.'-'. '$sec.'">
123
124             </div>
125
126             <br/>
127
128             <div class="input-group">
129                 <span
... class="input-group-addon">Altura</span>
130
131                 <input type="number" step="any"

```

```

131... name="altTrapezio'.$tram.'-'. $sec.'" class=" form-control datosSeccion"
... aria-describedby="altTrapezio'.$tram.'-'. $sec.'" placeholder="metros" min="0"
... id="altTrapezio'.$tram.'-'. $sec.'">
132         </div>
133
134         <br/>
135
136         <div class="input-group">
137             <span class="input-group-addon">Longitud de
... la base</span>
138
139                 <input type="number" step="any"
... name="baseTrapezio'.$tram.'-'. $sec.'" class=" form-control datosSeccion"
... aria-describedby="baseTrapezio'.$tram.'-'. $sec.'" placeholder="metros" min="0"
... id="baseTrapezio'.$tram.'-'. $sec.'">
140         </div>';
141         if ($tram != $numTramos) {
142             echo '
143         <br/>
144         <p class="bg-info text-center">Continúe
... definiendo el tramo ' .($tram+1).'</p>';
145             }
146             echo '
147         </div>
148 <!--//////////////////////////////////TRIANGULAR//////////////////////////////////-->
149
150         <div id="triangular'.$tram.'-'. $sec.'"
... style="display: none;" class="col-xs-11">
151
152         <br/>
153
154         <div class="input-group">
155             <span class="input-group-addon">Longitud
... del borde superior del triángulo</span>
156
157                 <input type="number" step="any"
... name="supTriangulo'.$tram.'-'. $sec.'" class=" form-control datosSeccion"
... aria-describedby="supTriangulo'.$tram.'-'. $sec.'" placeholder="metros" min="0"
... id="supTriangulo'.$tram.'-'. $sec.'">
158         </div>
159
160         <br/>
161
162         <div class="input-group">
163             <span class="input-group-addon">Altura del
... triángulo</span>
164
165                 <input type="number" step="any"
... name="altTriangulo'.$tram.'-'. $sec.'" class=" form-control datosSeccion"
... aria-describedby="altTriangulo'.$tram.'-'. $sec.'" placeholder="metros" min="0"
... id="altTriangulo'.$tram.'-'. $sec.'">
166

```

```

167         </div>';
168         if ($tram != $numTramos) {
169             echo '
170             <br/>
171             <p class="bg-info text-center">Continúe
... definiendo el tramo ' . ($tram+1) . '</p>';
172         }
173         echo '
174     </div>
175
176     <div class="col-xs-11">
177
178         <br/>
179         <div class="input-group">
180
181             <span class="input-group-addon
... letra">Número de Manning del tramo '$tram.'</span>
182
183             <input class="form-control col-xs-2
... input-manning" aria-describedby="Manning'$tram.'" type="number" step="any"
... id="Manning'$tram.'" name="Manning'$tram.'" required="required"
... autocomplete="on" value="0.01" min="0" />
184         </div>
185     </div>
186 </div>
187
188     <div class="col-xs-6">
189         <div id="container'$tram.'"
... class="col-xs-11"></div>
190         <div class="col-xs-1"></div>
191     </div>
192
193
194     <input class="col-xs-2" type="number"
... name="seccion'$tram.'" required="required" autocomplete="on" value="1" min="1"
... hidden/>
195     ';
196     } //cierra sec
197     if ($tram == $numTramos) {
198         echo '
199         <br/>
200         <br/>
201         <input class="btn boton-siguiente boton-letra" type="submit"
... value="Siguiente &rarr;" />
202         </div>'; //Este div es el de la ultima parte, con el botón de
... submit.
203     }else{
204         echo '
205         </div>'; //Este div es el que mete cada parte que se muestra.
206     } //cierra tram
207     if (isset($_POST['editTramos'])) {
208         echo '

```

```

209         <div id="datosEdit" hidden>
210             <input type="number" id="seccionesEdit" name="seccionesEdit"
... step="any" value="'.$_POST['editTramos'].'"' hidden>
211             <input type="number" id="numTramosEdit" name="numTramosEdit"
... value="'.$_POST['numTramosEdit'].'"' hidden>
212             <input type="text" id="carpetaEdit" name="carpetaEdit"
... value="'.$_POST['carpetaEdit'].'"' hidden>
213         </div>
214         ';
215     }
216     echo'
217         <input name="numTramos" id="numTramos" type="number"
... required="required" value="'.$_numTramos.'" hidden/>
218         <input name="nombreCaso" type="text" required="required"
... value="'.$_nombreCaso.'" hidden/>
219
220     </form>
221 </div>
222 ';
223 ?>
224 </body>
225
226 </html>

```

## Javascript para la paginación (utilizado en secciones, topología y condiciones de contorno)

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 function prueba(identificador, numTramos) {
4     for (i = 1; i <= numTramos; i++) {
5         $("#tramo" + i).hide();
6         if (i == identificador) {
7             $("#tramo" + identificador).show();
8         }
9     };
10 };
11
12 function paginarSalida(identificador, numTramos) {
13     for (i = 1; i <= numTramos; i++) {
14         $("#tramoSalida" + i).hide();
15         if (i == identificador) {
16             $("#tramoSalida" + identificador).show();
17         }
18     }
19 };
20
21 function refreshCurrentViewTramo(tramo, cursec) {
22
23     switch (parseInt($("#selecttipo" + tramo + "-" + cursec).val())) {
24         case 1:
25             $("#semicircular" + tramo + "-" + cursec).show();
26             $("#trapezoidal" + tramo + "-" + cursec).hide();
27             $("#triangular" + tramo + "-" + cursec).hide();
28             $("#otra" + tramo + "-" + cursec).hide();
29
30             $("#radio" + tramo + "-" + cursec).focus();
31
32             break;
33
34         case 2:
35             $("#semicircular" + tramo + "-" + cursec).hide();
36             $("#trapezoidal" + tramo + "-" + cursec).show();
37             $("#triangular" + tramo + "-" + cursec).hide();
38             $("#otra" + tramo + "-" + cursec).hide();
39
40             $("#supTrapezio" + tramo + "-" + cursec).focus();
41
42             break;
43
44         case 3:
45             $("#semicircular" + tramo + "-" + cursec).hide();
46             $("#trapezoidal" + tramo + "-" + cursec).hide();
47             $("#triangular" + tramo + "-" + cursec).show();
48             $("#otra" + tramo + "-" + cursec).hide();
49
50             $("#supTriangulo" + tramo + "-" + cursec).focus();
51
52             break;
```

```
53
54     default:
55         $("#semicircular" + tramo + "-" + cursec).hide();
56         $("#trapezoidal" + tramo + "-" + cursec).hide();
57         $("#triangular" + tramo + "-" + cursec).hide();
58         $("#otra" + tramo + "-" + cursec).hide();
59     }
60 };
61
62 $(".input-manning").keypress(function qw() {
63     console.log("entroooo");
64     //9 es el código de la tecla tab
65     if (e.which == 9) {
66         alert('Has pulsado enter!');
67     }
68
69 });
```

## Javascript secciones

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 $(document).ready(function() {
4     function editSecciones() {
5
6         if ($("#seccionesEdit").length > 0) {
7             //En esta funcion hay que coger los Manning de todos los tramos,
... los tipos de secciones y sus datos.
8             carpeta = $("#carpetaEdit").val();
9             numTramos = parseInt($("#numTramosEdit").val());
10
11             $.ajax({
12                 url: 'cloud/' + carpeta + '/sections.input',
13                 type: "get",
14                 async: false,
15                 success: function(data) {
16                     // Separa las lineas del archivo
17                     var lineasSections = data.split('\n');
18                     i = 0;
19
20                     $.ajax({
21                         url: 'cloud/' + carpeta + '/geometryXYZ.input',
22                         type: "get",
23                         async: false,
24                         success: function(dato) {
25                             var lineasManning = dato.split('\n');
26                             j = 0;
27
28                             for (var tram = 1; tram <= numTramos; tram++) {
29
30                                 itemSections = lineasSections[i].split(" ");
31                                 tipoSeccion = parseInt(itemSections[0]);
32
33                                 itemManning = lineasManning[j + 1].split(" ");
34                                 valorManning = parseFloat(itemManning[1]);
35                                 $("#Manning" + tram).val(valorManning);
36
37                                 $("#selecttipo" + tram + '-1').val(tipoSeccion);
38
39                                 switch (tipoSeccion) {
40                                     case 1:
41                                         itemSections = lineasSections[i +
... 1].split(" ");
42                                         radio = parseFloat(itemSections[0]);
43
44                                         $("#radio" + tram + '-1').val(radio);
45                                         i = i + 2;
46                                         j = j + 24;
47                                         break;
48
49                                     case 2:
50                                         lineaSupTrapezio = lineasSections[i +
```

```

50... 1].split(" ");
51                                     $("#supTrapecio" + tramo +
... '-1').val(lineaSupTrapecio[0]);
52                                     lineaAltTrapecio = lineasSections[i +
... 2].split(" ");
53                                     $("#altTrapecio" + tramo +
... '-1').val(lineaAltTrapecio[0]);
54                                     lineaBaseTrapecio = lineasSections[i +
... 3].split(" ");
55                                     $("#baseTrapecio" + tramo +
... '-1').val(lineaBaseTrapecio[0]);
56
57                                     i = i + 4;
58                                     j = j + 16;
59                                     break;
60
61                                     case 3:
62                                     lineaSupTriangulo = lineasSections[i +
... 1].split(" ");
63                                     $("#supTriangulo" + tramo +
... '-1').val(lineaSupTriangulo[0]);
64                                     lineaAltTriangulo = lineasSections[i +
... 2].split(" ");
65                                     $("#altTriangulo" + tramo +
... '-1').val(lineaAltTriangulo[0]);
66                                     i = i + 3;
67                                     j = j + 12;
68                                     break;
69                                     }
70                                 }
71                            }
72                        });
73                    }
74                });
75            }
76        }
77
78        function mostrarRellenos() {
79            cursec = 1;
80            for (var tramo = 1; tramo <= parseInt($("#numTramos").val()); tramo++)
81            {
82                if ($("#radio" + tramo + "-" + cursec).val() != "") {
83                    $("#semicircular" + tramo + "-" + cursec).show();
84                    $("#trapezoidal" + tramo + "-" + cursec).hide();
85                    $("#triangular" + tramo + "-" + cursec).hide();
86                    $("#otra" + tramo + "-" + cursec).hide();
87                } else if (($("#baseTrapecio" + tramo + "-" + cursec).val() != "")
88                || ($("#altTrapecio" + tramo + "-" + cursec).val() != "") || ($("#supTrapecio"
89                + tramo + "-" + cursec).val() != "")) {
87                    $("#semicircular" + tramo + "-" + cursec).hide();
88                    $("#trapezoidal" + tramo + "-" + cursec).show();
89                    $("#triangular" + tramo + "-" + cursec).hide();

```

```

90         $("#otra" + tramo + "-" + cursec).hide();
91     } else if (($("#supTriangulo" + tramo + "-" + cursec).val() != ""))
... || ($("#altTriangulo" + tramo + "-" + cursec).val() != "") {
92         $("#semicircular" + tramo + "-" + cursec).hide();
93         $("#trapezoidal" + tramo + "-" + cursec).hide();
94         $("#triangular" + tramo + "-" + cursec).show();
95         $("#otra" + tramo + "-" + cursec).hide();
96     }
97 }
98 };
99
100 editSecciones();
101 mostrarRellenos();
102
103 function opcionesGrafica(tramo, JSON) {
104
105     var options = {
106         chart: {
107             style: {
108                 fontFamily: 'Open Sans Condensed'
109             },
110             renderTo: 'container' + tramo,
111             type: 'line'
112         },
113         title: {
114             text: 'Sección del tramo ' + tramo
115         },
116         xAxis: {
117             title: {
118                 text: 'X'
119             },
120             minPadding: 0.1,
121             maxPadding: 0.1,
122             maxZoom: 2.5,
123             plotLines: [{
124                 value: 0,
125                 width: 0.5,
126                 color: '#cccccc'
127             }]
128         },
129         yAxis: {
130             title: {
131                 text: 'Y'
132             },
133             minPadding: 0.1,
134             maxPadding: 0.1,
135             maxZoom: 2.5,
136             plotLines: [{
137                 value: 0,
138                 width: 0.5,
139                 color: '#cccccc'
140             }]

```

```

141     },
142     legend: {
143         enabled: false,
144         layout: 'vertical',
145         align: 'center',
146         verticalAlign: 'bottom',
147         borderWidth: 0
148     },
149     tooltip: {
150         formatter: function() {
151             return 'X: ' + this.x + '; Y: ' + this.y;
152         }
153     },
154     series: []
155 };
156
157 options.series = JSON;
158
159 // Se crea el gráfico
160
161 var chart = new Highcharts.Chart(options);
162 }
163
164 function graficasSecciones() {
165
166     for (var tramo = 1; tramo <= parseInt($("#numTramos").val()); tramo++)
167     {
168
169         switch (parseInt($("#selecttipo" + tramo + "-1").val())) {
170             case 1:
171                 radio = parseFloat($("#radio" + tramo + "-1").val());
172
173                 if ($("#radio" + tramo + "-1").val() != "") {
174                     grados = 180;
175                     radianes = grados * Math.PI / 180;
176                     quesitos = 8;
177                     for (var p = 0; p <= quesitos; p++) {
178                         angulo = radianes + p * (radianes / quesitos);
179                         if (p <= quesitos * 0.5) {
180                             window["punto" + p] = [parseFloat((radio -
181 ... radio * (Math.abs(Math.cos(angulo)))).toFixed(4)), parseFloat((radio - radio *
182 ... (Math.abs(Math.sin(angulo)))).toFixed(4))];
183                         } else {
184                             window["punto" + p] = [parseFloat(((2 * radio)
185 ... - radio * (1 - Math.abs(Math.cos(angulo)))).toFixed(4)), parseFloat((radio -
186 ... radio * (Math.abs(Math.sin(angulo)))).toFixed(4))];
187                         }
188                     }
189                 }
190                 //son ocho puntos, vamos a sacarlos en orden para
191 ... dibujar el poligono
192                 for (var k = 0; k <= quesitos; k++) {

```

```

187
188         if (k == 0) {
189             poligono = [];
190         }
191         poligono.push(window['punto' + k]);
192     }
193
194     tram = 'semicírculo tramo ' + tramo;
195     grosor = 5;
196
197     serie = {
198         name: tram,
199         data: poligono,
200         lineWidth: grosor
201     };
202
203     var JSON = [];
204
205     JSON.push(serie);
206     opcionesGrafica(tramo, JSON);
207 }
208 break;
209
210 case 2:
211
212     if ($("#baseTrapezio" + tramo + "-1").val() != '') {
213         if ($("#altTrapezio" + tramo + "-1").val() != '') {
214             if ($("#supTrapezio" + tramo + "-1").val() != '') {
215
216                 baseTrapezio = parseFloat($("#baseTrapezio" +
... tramo + "-1").val());
217                 altTrapezio = parseFloat($("#altTrapezio" +
... tramo + "-1").val());
218                 supTrapezio = parseFloat($("#supTrapezio" +
... tramo + "-1").val());
219                 //son cuatro puntos, vamos a sacarlos en orden
... para dibujar el poligono
220                 puntoSupIz = [0, altTrapezio];
221                 puntoInfIz = [0.5 * (supTrapezio -
... baseTrapezio), 0];
222                 puntoInfDcho = [0.5 * (supTrapezio -
... baseTrapezio) + baseTrapezio, 0];
223                 puntoSupDcho = [supTrapezio, altTrapezio];
224
225                 poligono = [puntoSupIz, puntoInfIz,
... puntoInfDcho, puntoSupDcho];
226                 tram = 'Trapezio tramo ' + tramo;
227                 grosor = 5;
228
229                 serie = {
230                     name: tram,
231                     data: poligono,

```

```

232         lineWidth: grosor
233     };
234
235     var JSON = [];
236
237
238     JSON.push(serie);
239
240     opcionesGrafica(tramo, JSON);
241 }
242 }
243 }
244 break;
245
246 case 3:
247
248     if ($("#supTriangulo" + tramo + "-1").val() != '') {
249         if ($("#altTriangulo" + tramo + "-1").val() != '') {
250
251             altTriangulo = parseFloat($("#altTriangulo" + tramo
... + "-1").val());
252             supTriangulo = parseFloat($("#supTriangulo" + tramo
... + "-1").val());
253             //son cuatro puntos, vamos a sacarlos en orden para
... dibujar el poligono
254             puntoSupIz = [0, altTriangulo];
255             puntoInf = [0.5 * (supTriangulo), 0];
256             puntoSupDcho = [supTriangulo, altTriangulo];
257
258             poligono = [puntoSupIz, puntoInf, puntoSupDcho];
259             tram = 'Triangulo tramo ' + tramo;
260             grosor = 5;
261
262             serie = {
263                 name: tram,
264                 data: poligono,
265                 lineWidth: grosor
266             };
267
268             var JSON = [];
269
270             JSON.push(serie);
271
272             opcionesGrafica(tramo, JSON);
273         }
274     }
275     break;
276 }
277 }
278 }
279 $("#input.datosSeccion").change(function() {
280     graficasSecciones();

```

```
281     })
282     $("#a#myLink").click(function() {
283         graficasSecciones();
284     })
285
286     $(document).ready(function() {
287         graficasSecciones();
288         mostrarRellenos();
289     });
290 });
```

## HTML

```

1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <meta charset="UTF-8">
8     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
9     <link rel="stylesheet" href="css/bootstrap.min.css">
10    <link rel="stylesheet" href="css/main.css">
11    <link href="fonts/googleFonts.woff2" rel="stylesheet">
12    <script src="js/jquery.min.js"></script>
13    <script src="js/bootstrap.min.js"></script>
14    <script src="js/highcharts.js"></script>
15    <script src="js/exporting.js"></script>
16    <script src="js/funcionesTopologia.js"></script>
17    <script src="js/paginacion.js"></script>
18
19    <title>Topología</title>
20 </head>
21
22 <body>
23
24     <div class="row">
25         <div class="col-xs-1"></div>
26         <h2 class="col-xs-4 paso-plantilla">TOPOLOGÍA DE LA RED</h2>
27         <h1 class="col-xs-6
... titulo-plantilla">N&nbsp;A&nbsp;V&nbsp;I&nbsp;G&nbsp;L&nbsp;I</h1>
28         <div class="col-xs-1"></div>
29     </div>
30
31     <?php
32
33     $nombreCaso = $_POST['nombreCaso'];
34     $numTramos = $_POST['numTramos'];
35
36
37     //El primer div es para centrar y colocar en un sitio específico y el ul es
... para el fondo de los botones de la paginación. El li es para los botones de
... paginación (pinta todos).
38     echo '
39
40
41     <div class="row">
42         <div class="col-xs-3 tamaño-letra alinear-derecha">Tramos_</div>
43         <div class="col-xs-7">
44             <ul class="pagination letra mediano">';
45             for ($tram=1;$tram<=$numTramos;$tram++){
46                 echo '
47
48                 <li><a id="myLink" href="#"

```

```

48... onclick="prueba('.$tram.', '.$numTramos.');"'.$tram.'</a></li>
49
50         ';
51     }
52     echo '
53     </ul>
54 </div>
55     <button type="button" class=" btn glyphicon glyphicon-info-sign btn-sm"
... data-toggle="modal" data-target="#info" title="¿Qué puntos son
... estos?"></button>
56
57
58         <div class="modal fade letra" id="info">
59         <div class="modal-dialog" role="document">
60         <div class="modal-content">
61         <div class="modal-header">
62         <button type="button" class="close"
... data-dismiss="modal" aria-label="Close">
63             <span aria-hidden="true">&times;</span>
64         </button>
65         <h4 class="modal-title" id="myModalLabel">¿Cuáles
... son estos puntos?</h4>
66         </div>
67         <div class="modal-body">
68             Estos puntos definen la parte
... inferior central de cada sección, es decir, el fondo del canal. En las
... siguientes imágenes se muestran dichos puntos de entrada y salida respecto a la
... secciones posibles.</br>
69             <IMG SRC="css/img/semicirculo.jpg" WIDTH=255
... HEIGHT=180>Semicírculo</IMG></br>
70             <IMG SRC="css/img/trapecio.jpg" WIDTH=255
... HEIGHT=180>Rectángulo/trapecio</IMG></br>
71             <IMG SRC="css/img/triangulo.jpg" WIDTH=255
... HEIGHT=180>Triángulo</IMG>
72             <!-- //2480 × 1748 -->
73
74         </div>
75         <div class="modal-footer">
76         <button type="button" class="btn btn-secondary"
... data-dismiss="modal">Aceptar</button>
77         <!-- <button type="button" class="btn
... btn-primary">Aceptar</button> -->
78         </div>
79     </div>
80 </div>
81 </div>
82 </div>';
83
84     echo '
85
86 <form class="row" action="condContorno.php" method="post"
... enctype="multipart/form-data" novalidate>

```

```

87     <div class="col-xs-1 letra"></div>
88     ';
89
90     for ($tram=1; $tram<=$numTramos; $tram++){
91         $Manning = $_POST['Manning'.$tram];
92
93
94         //Esto hace que se muestre la primera vez los elementos del
... tramo 1.
95         if($tram==1){
96             echo '
97             <div class="col-xs-4 letra" id="tramo'.$tram.'"
... style="display:block;">';
98             }else{
99             echo '
100            <div class="col-xs-4 letra" id="tramo'.$tram.'"
... style="display:none;">';
101            }
102
103            $numSec = $_POST['seccion'.$tram];
104
105            for ($sec=1; $sec<=$numSec; $sec++){
106
107                $forma = $_POST['forma'.$tram.'-'.$sec];
108
109                echo '<input name="forma'.$tram.'-'.$sec.'" type="number"
... required="required" value="'.$forma.'" hidden/>
110                <input name="Manning'.$tram.'" type="number"
... required="required" value="'.$Manning.'" hidden/>';
111
112                switch ($forma){
113
114                    case 1:
115                        $radio = $_POST['radio'.$tram.'-'.$sec];
116
117                        echo '<input name="radio'.$tram.'-'.$sec.'"
... type="number" step="any" required="required" value="'.$radio.'" hidden/>';
118                        break;
119
120                    case 2:
121                        $sup2 = $_POST['supTrapecio'.$tram.'-'.$sec];
122                        $alt2 = $_POST['altTrapecio'.$tram.'-'.$sec];
123                        $base = $_POST['baseTrapecio'.$tram.'-'.$sec];
124
125                        echo '
126                        <input name="baseTrapecio'.$tram.'-'.$sec.'"
... type="number" step="any" required="required" value="'.$base.'" hidden/>
127
128                        <input name="altTrapecio'.$tram.'-'.$sec.'"
... type="number" step="any" required="required" value="'.$alt2.'" hidden/>
129
130                        <input name="supTrapecio'.$tram.'-'.$sec.'"

```

```

130... type="number" step="any" required="required" value="'.$sup2.'" hidden/>';
131         break;
132
133         case 3:
134             $sup3 = $_POST['supTriangulo'].$tram.'-'. $sec];
135             $alt3 = $_POST['altTriangulo'].$tram.'-'. $sec];
136             echo '
137
138                 <input name="supTriangulo'.$tram.'-'. $sec.'"
... type="number" step="any" required="required" value="'. $sup3.'" hidden/>>
139
140                 <input name="altTriangulo'.$tram.'-'. $sec.'"
... type="number" step="any" required="required" value="'. $alt3.'" hidden/>>';
141         break;
142
143         default:
144             echo '<b><FONT COLOR="red">No has seleccionado una
... forma de la seccion del tramo '.$tram.'</FONT></b>
145                 <br/>';
146         }
147
148     //////////////////////////////////////
149     //////////////////////////////////////EMPIEZA AQUI EL FORMULARIO QUE SE
150     VE////////////////////////////////////
151
152     //////////////////////////////////////
153     echo'
154
155         <b class="letra">¿Qué coordenadas en planta tiene la
... entrada del tramo '.$tram.'?</b>
156         <div >
157
158             <br/>
159
160             <div class="input-group col-xs-12">
161                 <span class="input-group-addon letra
... letra">X</span>
162
163                 <input type="number" step="any"
... name="entradaX'.$tram.'" class="form-control pierdeFoco"
... aria-describedby="entradaX'.$tram.'" placeholder="metros" value="1"
... id="entradaX'.$tram.'">
164
165             </div>
166
167         </div>
168
169         <br/>
170
171         <div>
172
173             <div class="col-xs-12 input-group">

```

```

171         <span class="input-group-addon letra" >Y</span>
172
173         <input type="number" step="any"
... name="entradaY'.'.$tram.'" class="col-xs-1 form-control pierdeFoco"
... aria-describedby="entradaY'.'.$tram.'" placeholder="metros" value="2"
... id="entradaY'.'.$tram.'">
174     </div>
175
176 </div>
177
178 <br/>
179
180 <div>
181
182     <div class="col-xs-12 input-group">
183         <span class="input-group-addon letra">Cota
... (Z)</span>
184
185         <input type="number" step="any"
... name="entradaZ'.'.$tram.'" class="col-xs-1 form-control pierdeFoco"
... aria-describedby="entradaZ'.'.$tram.'" placeholder="metros" value="10"
... id="entradaZ'.'.$tram.'">
186     </div>
187
188 </div>
189
190
191
192         <br/><br/>
193
194
195
196         <b class="letra">¿Qué coordenadas en planta tiene la salida
... del tramo '.'.$tram.'?</b>
197
198         <div >
199
200             <br/>
201
202             <div class="col-xs-12 input-group">
203                 <span class="input-group-addon letra">X</span>
204
205                 <input type="number" step="any"
... name="salidaX'.'.$tram.'" class="col-xs-1 form-control pierdeFoco"
... aria-describedby="salidaX'.'.$tram.'" placeholder="metros" value="6"
... id="salidaX'.'.$tram.'">
206             </div>
207
208         </div>
209
210         <br/>
211

```

```

212         <div>
213
214             <div class="col-xs-12 input-group">
215                 <span class="input-group-addon letra">Y</span>
216
217                     <input type="number" step="any"
218 ... name="salidaY'.$tram.'" class="col-xs-1 form-control pierdeFoco"
219 ... aria-describedby="salidaY'.$tram.'" placeholder="metros" value="4"
220 ... id="salidaY'.$tram.'">
221             </div>
222
223         <br/>
224
225         <div>
226
227             <div class="col-xs-12 input-group">
228                 <span class="input-group-addon letra">Cota
229 ... (Z)</span>
230
231                 <input type="number" step="any"
232 ... name="salidaZ'.$tram.'" class="col-xs-1 form-control pierdeFoco"
233 ... aria-describedby="salidaZ'.$tram.'" placeholder="metros" value="9"
234 ... id="salidaZ'.$tram.'">
235             </div>
236
237         </div>
238         <br/>
239
240         <div class="col-xs-12 row input-group letra">
241
242             <button type="button" class="btn glyphicon
243 ... glyphicon-plus col-xs-1 btn-sm" data-toggle="modal"
244 ... data-target="#myModal'.$tram.'" title="Calado mínimo & Celdas de
245 ... cálculo"></button>
246
247             <div>
248                 <div class="col-xs-2"></div>
249                 <input type="checkbox" title="Marca esta caja si es
250 ... entrada de la red" name="entrada'.$tram.'" id="entrada'.$tram.'" value=""> El
251 ... tramo '$tram.'" es entrada de la red.<br>
252             </div>
253
254         </div>
255
256         <div class="modal fade" id="myModal'.$tram.'">

```

```

252         <div class="modal-dialog" role="document">
253             <div class="modal-content">
254                 <div class="modal-header">
255                     <button type="button" class="close"
... data-dismiss="modal" aria-label="Close">
256                         <span aria-hidden="true">&times;</span>
257                     </button>
258                     <h4 class="modal-title"
... id="myModallabel">Configuración avanzada</h4>
259                 </div>
260                 <div class="modal-body">
261
262
263
264                     <br/>
265
266                     <div class="row">
267
268                         <br/>
269                         <div class="col-xs-3"></div>
270                         <div class="input-group
... col-xs-6">
271
272                             <span
... class="input-group-addon letra">Calado inicial del tramo '.$tram.' [m]</span>
273
274                             <input class="form-control
... calados" type="number" name="hInicial'.$tram.'" required="required" step="any"
... autocomplete="on" value="0.2" min="0" id="hInicial'.$tram.'"/>
275
276                             </div>
277                             <div class="col-xs-3"></div>
278                         </div>
279
280                     <br/>
281
282                     <div class="row">
283
284                         <br/>
285                         <div class="col-xs-3"></div>
286                         <div class="input-group
... col-xs-6">
287
288                             <span
... class="input-group-addon letra">Calado mínimo del tramo '.$tram.' [m]</span>
289
290                             <input class="form-control
... calados" aria-describedby="calado'.$tram.'" type="number" step="any"
... name="calado'.$tram.'" required="required" autocomplete="on" value="0.001"
... min="0.000001" id="calado'.$tram.'"/>
291
292                             </div>
293                             <div class="col-xs-3"></div>

```

```

293         </div>
294
295         <br/>
296
297         <div class="row">
298
299             <br/>
300             <div class="col-xs-3"></div>
301             <div class="input-group
... col-xs-6">
302
303                 <span
... class="input-group-addon letra" >Número de celdas de cálculo del tramo
... '.$tram.'</span>
304
305                 <input class="form-control"
... type="number" name="NCELLS'.$tram.'" id="NCELLS'.$tram.'" required="required"
... autocomplete="on" value="50" min="0" />
306
307             </div>
308             <div class="col-xs-3"></div>
309         </div>
310
311
312         </div>
313         <div class="modal-footer">
314             <button type="button" class="btn btn-secondary"
... data-dismiss="modal">Aceptar</button>
315             <!-- <button type="button" class="btn
... btn-primary">Aceptar</button> -->
316             </div>
317         </div>
318     </div>
319 </div>
320
321     ';//cierra sec
322
323     if ($tram == $numTramos && $forma !=0) { //Solo lo muestro en
... el ultimo tramo y si no hay errores
324     echo '
325     <div class="col-xs-7"></div>
326     <input class="btn boton-letra" type="submit" value="Siguiente
... &rarr;" />
327     </div>
328
329     ';//Este div es el de la ultima parte, con el botón de submit.
330     }else{
331     echo '
332     <br/>
333     <p class="bg-info text-center">Continúe definiendo el tramo
... '.$(tram+1).'

```

```

335         ';//Este div es el que mete cada parte que se muestra.
336     } //cierra tram
337
338     if (isset($_POST['seccionesEdit'])&&($_POST['seccionesEdit']==1) {
339         echo '
340             <div id="datosEdit" hidden>
341                 <input type="number" id="topologiaEdit" name="topologiaEdit"
... step="any" value="'. $_POST['seccionesEdit']. '" hidden>
342                 <input type="number" id="numTramosEdit" name="numTramosEdit"
... value="'. $_POST['numTramosEdit']. '" hidden>
343                 <input type="text" id="carpetaEdit" name="carpetaEdit"
... value="'. $_POST['carpetaEdit']. '" hidden>
344             </div>
345         ';
346     }
347     echo '
348         <div class="col-xs-7">
349             <div id="container" class="col-xs-10" style="min-width:
... 310px; height: 250px; margin: 0 auto"></div>
350             <div id="container2" class="col-xs-10" style="min-width:
... 310px; height: 250px; margin: 0 auto"></div>
351         </div>
352
353         <input name="numTramos" id="numTramos" type="number"
... required="required" value="'. $numTramos. '" hidden/>
354         <input name="nombreCaso" type="text" required="required"
... value="'. $nombreCaso. '" hidden/>
355         <input name="seccion'. $tram. '" type="number"
... required="required" value="'. $numSec. '" hidden/>
356 </form>
357 ';
358 ?>
359 </body>
360
361 </html>

```

## Javascript

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 $(document).ready(function() {
4     $("input#entradaX1").focus();
5
6     editTopologia();
7     graficaPlanta();
8     graficaAlzado();
9
10
11     function editTopologia() {
12
13         if ($("#topologiaEdit").length > 0) {
14             carpeta = $("#carpetaEdit").val();
15             numTramos = parseInt($("#numTramosEdit").val());
16             $.ajax({
17                 url: 'cloud/' + carpeta + '/entries.input',
18                 type: "get",
19                 async: false,
20                 success: function(datoEntries) {
21
22                     var lineasEntries = datoEntries.split('\n');
23                     cuentaLineasEntries = lineasEntries.length;
24                     for (var i = 0; i < cuentaLineasEntries; i++) {
25                         tramoEntrada = parseInt(lineasEntries[i]);
26                         $("input#entrada" + tramoEntrada).attr('checked',
27 ... "checked");
28                     }
29                 });
30             $.ajax({
31                 url: 'cloud/' + carpeta + '/initial.input',
32                 type: "get",
33                 async: false,
34                 success: function(datoInitial) {
35                     $.ajax({
36                         url: 'cloud/' + carpeta + '/geometryXYZ.input',
37                         type: "get",
38                         async: false,
39                         success: function(datoGeometry) {
40
41                             var lineasInitial = datoInitial.split('\n');
42
43                             var lineasGeometry = datoGeometry.split('\n');
44                             i = 0;
45                             for (var tram = 1; tram <= numTramos; tram++) {
46
47                                 tipoInitialLinea = lineasInitial[tram -
48 ... 1].split(" ");
49
50                                 $("input#hInicial" +
51 ... tram).val(parseFloat(tipoInitialLinea[3]));
```

```

50
51         tipoSeccionLinea = lineasGeometry[i +
... 4].split(" ");
52         tipoSeccion = parseInt(tipoSeccionLinea[1]);
53
54         lineaCaladoMin = lineasGeometry[i + 2].split("
... ");
55         $("input#calado" +
... tram).val(parseFloat(lineaCaladoMin[1]));
56
57         lineaNcells = lineasGeometry[i].split(" ");
58         $("input#NCELLS" +
... tram).val(parseFloat(lineaNcells[1]));
59
60         switch (tipoSeccion) {
61
62             case 9:
63                 coordenadasEntradaLinea =
... lineasGeometry[i + 9].split(" ");
64                 coordenadasSalidaLinea =
... lineasGeometry[i + 19].split(" ");
65                 i = i + 24;
66
67                 break;
68
69             case 5:
70                 coordenadasEntradaLinea =
... lineasGeometry[i + 7].split(" ");
71                 coordenadasSalidaLinea =
... lineasGeometry[i + 13].split(" ");
72                 i = i + 16;
73
74                 break;
75
76             case 3:
77                 coordenadasEntradaLinea =
... lineasGeometry[i + 6].split(" ");
78                 coordenadasSalidaLinea =
... lineasGeometry[i + 10].split(" ");
79                 i = i + 12;
80
81                 break;
82         }
83         $("input#entradaX" +
... tram).val(parseFloat(coordenadasEntradaLinea[0]));
84         $("input#entradaY" +
... tram).val(parseFloat(coordenadasEntradaLinea[1]));
85         $("input#entradaZ" +
... tram).val(parseFloat(coordenadasEntradaLinea[2]));
86
87
88         $("input#salidaX" +

```

```

88... tram).val(parseFloat(coordenadasSalidaLinea[0]));
89                                     $("input#salidaY" +
... tram).val(parseFloat(coordenadasSalidaLinea[1]));
90                                     $("input#salidaZ" +
... tram).val(parseFloat(coordenadasSalidaLinea[2]));
91                                     }
92                                 }
93                            })
94                        }
95                    })
96                }
97            }
98
99        function graficaPlanta() {
100
101            for (var i = 1; i <= parseInt($('#numTramos').val()); i++) {
102
103                puntosEntrada = [parseFloat($('#input#entradaX' + i).val()),
... parseFloat($('#input#entradaY' + i).val())];
104                puntosSalida = [parseFloat($('#input#salidaX' + i).val()),
... parseFloat($('#input#salidaY' + i).val())];
105
106                puntosTramo = [puntosEntrada, puntosSalida];
107
108                tramo = 'Tramo' + i;
109
110                grosor = 5.5;
111
112                serie = {
113                    name: tramo,
114                    data: puntosTramo,
115                    lineWidth: grosor
116                };
117
118                if (i == 1) {
119                    var JSON = [];
120                }
121                JSON.push(serie);
122            }
123
124            var options = {
125                chart: {
126                    style: {
127                        fontFamily: 'Open Sans Condensed'
128                    },
129                    renderTo: 'container',
130                    type: 'line'
131                },
132                title: {
133                    text: 'Planta de la red'
134                },
135                tooltip: {

```

```

136         formatter: function() {
137             return 'X: ' + this.x + '; Y: ' + this.y;
138         }
139     },
140     legend: {
141         layout: 'vertical',
142         align: 'right',
143         verticalAlign: 'middle',
144         borderWidth: 0
145     },
146     xAxis: {
147         title: {
148             text: 'X'
149         }
150     },
151     yAxis: {
152         title: {
153             text: 'Y'
154         }
155     },
156     series: []
157 };
158
159 options.series = JSON;
160
161 // Se crea el gráfico
162 var chart = new Highcharts.Chart(options);
163 };
164
165 function graficaAlzado() {
166
167     for (var i = 1; i <= parseInt($('#numTramos').val()); i++) {
168
169         puntosEntrada = [parseFloat($('#input#entradaX' + i).val()),
170 ... parseFloat($('#input#entradaZ' + i).val())];
171         puntosSalida = [parseFloat($('#input#salidaX' + i).val()),
172 ... parseFloat($('#input#salidaZ' + i).val())];
173
174         puntosTramo = [puntosEntrada, puntosSalida];
175
176         tramo = 'Tramo' + i;
177
178         grosor = 5.5;
179
180         serie = {
181             name: tramo,
182             data: puntosTramo,
183             lineWidth: grosor
184         };
185
186         if (i == 1) {
187             var JSON = [];

```

```

186     }
187
188     JSON.push(serie);
189 }
190
191 var options = {
192     chart: {
193         style: {
194             fontFamily: 'Open Sans Condensed'
195         },
196         renderTo: 'container2',
197         type: 'line'
198     },
199     title: {
200         text: 'Alzado de la red',
201     },
202     xAxis: {
203         title: {
204             text: 'X'
205         }
206     },
207     yAxis: {
208         title: {
209             text: 'Cota (Z)'
210         }
211     },
212     tooltip: {
213         formatter: function() {
214             return 'X: ' + this.x + '; Z: ' + this.y;
215         }
216     },
217     legend: {
218         layout: 'vertical',
219         align: 'right',
220         verticalAlign: 'middle',
221         borderWidth: 0
222     },
223     series: []
224 };
225
226 options.series = JSON;
227
228 // Se crea el gráfico
229 var chart = new Highcharts.Chart(options);
230 };
231
232 function comprobacionCalados() {
233     for (tram = 1; tram <= parseInt($("#numTramos").val()); tram++) {
234         $("#calado" + tram).attr("max", parseFloat($("#hInicial" +
... tram).val()));
235         $("#hInicial" + tram).attr("min", parseFloat($("#calado" +
... tram).val()));

```

```

236     }
237 }
238
239 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
240 ///////////////////////////////////////////////////////////////////Esto es cuando se actualizan los inputs////////////////////////////////////////////////////////////////
241 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
242
243 $(".pierdeFoco").change(function() {
244     graficaPlanta();
245 });
246
247 $(".pierdeFoco").change(function() {
248     graficaAlzado();
249 });
250
251 $("input.calados").change(function() {
252     comprobacionCalados();
253 });
254
255 });

```

## Condiciones de contorno

En este HTML se realizan todos los cálculos geométricos y de las uniones.

## HTML

```
1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <meta charset="UTF-8">
8     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
9     <link rel="stylesheet" href="css/bootstrap.min.css">
10    <link rel="stylesheet" href="css/main.css">
11    <script src="js/jquery.min.js"></script>
12    <script src="js/highcharts.js"></script>
13    <script src="js/exporting.js"></script>
14    <script src="js/bootstrap.min.js"></script>
15    <script src="js/paginacion.js"></script>
16    <script src="js/funcionesCC.js"></script>
17    <link href="fonts/googleFonts.woff2" rel="stylesheet">
18    <title>Condiciones de contorno</title>
19 </head>
20
21 <body>
22     <div class="row">
23         <div class="col-xs-1"></div>
24         <h2 class="col-xs-4 paso-plantilla">CONDICIONES DE CONTORNO</h2>
25         <h1 class="col-xs-6
... titulo-plantilla">N&nbsp;A&nbsp;V&nbsp;I&nbsp;G&nbsp;L&nbsp;I</h1>
26         <div class="col-xs-1"></div>
27     </div>
28     <?php
29     if
... (isset($_POST['topologiaEdit'])&&($_POST['carpetaEdit'])==$_POST['nombreCaso']
... ) {
30     $carpeta = $_POST['carpetaEdit'];
31     $rutaCarpeta = 'cloud/'.$carpeta;
32     $rutaArchivos = $rutaCarpeta."/";
33     unlink($rutaArchivos."geometryXYZ.input");
34     unlink($rutaArchivos."junctions.input");
35     unlink($rutaArchivos."entries.input");
36     unlink($rutaArchivos."sections.input");
37     unlink($rutaArchivos."initial.input");
38
39 }else{
40 $nombreCaso = $_POST['nombreCaso'];
41 $carpeta = uniqid()."_" . $nombreCaso;
42 $rutaCarpeta = 'cloud/'.$carpeta;
43 $rutaArchivos = $rutaCarpeta."/";
44 mkdir($rutaCarpeta);
45 }
46
47 //Esta ruta de carpetas es específica para cada ordenador
48
```

```

49 $numTramos = $_POST['numTramos'];
50
51
52 //Aqui simplemente se hace un array con los tramos que son entrada para poder
... paginar
53 //bien (que se muestre abierta al ppio el primer tramo de entrada) si no, se
... mostraría
54 //siempre el primer tramo aunque no fuese de entrada.
55 $tramosEntrada = array();
56 for ($stram=1; $stram <= $numTramos ; $stram++) {
57     if (isset($_POST['entrada'].$stram)) {
58         array_push($tramosEntrada, $stram);
59     }
60 }
61
62 //El primer div es para centrar y colocar en un sitio especifico y el ul es
... para el fondo de los botones de la paginacion. El li es para los botones de
... paginación (pinta todos).
63 echo '
64
65 <div class="row">
66     <div class="col-xs-10 sinPadding" align="center">
67         <ul class="pagination letra mediano sinPadding">';
68
69         echo '
70
71             <li class="sinPadding"><a id="clickEntrada" href="#"
... onclick="condicionContEnt();">ENTRADA</a></li>
72             <li class="sinPadding"><a id="clickSalida" href="#"
... onclick="condicionContSal();">SALIDA</a></li>
73
74             ';
75         echo '
76     </ul>
77 </div>
78 </div>
79
80 <form class="panel-body sinPadding" action="parametros.php" method="post"
... accept-charset="utf-8">
81
82 <input name="rutaArchivos" type="text" required="required"
... value="'. $rutaArchivos. '" hidden/>
83
84 <input name="carpeta" type="text" required="required" value="'. $carpeta. '"
... hidden/>
85
86     <div id="Entrada">
87
88         <div class="row">
89             <div class="col-xs-3 tamaño-letra alineado-derecha" >Tramos
... Entrada_</div>
90             <div class="col-xs-7 sinPadding" >

```

```

91         <ul class="pagination letra mediano">';
92         for ($tram=1;$tram<=$numTramos;$tram++){
93
94             if(isset($_POST['entrada'].$tram)){
95                 echo '
96
97                     <li><a id="myLink" href="#"
... onclick="prueba('.$tram.','.$numTramos.');" ">'.$tram.</a></li>
98
99                 ';
100             }
101         }
102         echo '
103     </ul>
104 </div>
105 </div>';
106
107     $esValido = false;
108
109     for ($tram=1; $tram<=$numTramos; $tram++){
110
111         //Esto hace que se muestre la primera vez los elementos del
... tramo 1.
112
113         if(isset($tramosEntrada[0]) && $tram==$tramosEntrada[0]){
114             $esValido = true;
115             echo '
116             <div id="tramo'.$tram.'" style="display:block;">';
117         }else{
118             echo '
119             <div id="tramo'.$tram.'" style="display:none;">';}
120
121
122         //Recojo los parametros que dependen del tramo en el
... que estés.
123             $numSec = 1;
124             $Manning = $_POST['Manning'].$tram];
125             $calado = $_POST['calado'].$tram];
126             $NCELLS = $_POST['NCELLS'].$tram];
127
128
129             //////////////////////////////////////
... aqui////////////////////////////////////
130
131             //////////////////////////////////////
132
133             echo '
134
135             <input type="number" id="entradaX'.$tram.'"
... name="entradaX'.$tram.'" step="any" value="'.$_POST['entradaX'].$tram.'"
... hidden>

```

```

135         <input type="number" id="entradaY".$tram.'"
... name="entradaY".$tram.'" step="any" value="'.$_POST['entradaY'.$tram].'"
... hidden>
136         <input type="number" id="salidaX".$tram.'"
... name="salidaX".$tram.'" step="any" value="'.$_POST['salidaX'.$tram].'"
... hidden>
137         <input type="number" id="salidaY".$tram.'"
... name="salidaY".$tram.'" step="any" value="'.$_POST['salidaY'.$tram].'"
... hidden>
138         <input type="number" id="Ncells".$tram.'"
... name="Ncells".$tram.'" step="any" value="'.$NCELLS.'" hidden>
139
140
141         <input name="seccion".$tram.'" type="number"
... required="required" value="'.$numSec.'" hidden/>
142         <input name="numTramos" type="number"
... required="required" value="'.$numTramos.'" hidden/>
143         <input name="rutaArchivos" type="text"
... required="required" value="'.$rutaArchivos.'" hidden/>
144         ';
145
146
147         for ($sec=1; $sec<=$numSec; $sec++){
148
149             $forma = $_POST['forma'.$tram.'-'.$sec];
150
151             echo '<input name="forma'.$tram.'-'.$sec.'"
... type="number" required="required" value="'.$forma.'" hidden/>';
152
153             switch ($forma){
154
155                 case 1://Semicirculo
156                     $numPuntos = 8;
157                     $radio =
... $_POST['radio'.$tram.'-'.$sec];
158
159                     $centroCirc = array
... ("x"=>$radio,"y"=>$radio);
160
161                     //Vamos a sacar 9 puntos, 8 partes, de
... definicion de la forma semicircular a 180
162                     $grados = 180;
163                     $numQuesitos = 8;
164                     $radianes = deg2rad($grados);
165
166                     for ($p=0; $p <= $numQuesitos; $p++) {
167                         if ($p<=(0.5*$numQuesitos)) {
168
... ${'punto'.$p.'.'tramo'.$tram.'.'seccion'.$sec} = array
... ("x"=>$radio*(1-abs(cos($p*$radianes/$numQuesitos+$radianes))), "y"=>$radio*(1-
... abs(sin($p*$radianes/$numQuesitos+$radianes))));
169

```

```

170                                     ${'distanciaEntradaPunto'}.${p} =
... $radio*abs(cos($p*$radianes/$numQuesitos+$radianes));
171                                     ${'distanciaEntradaPunto'}.${p} =
... round(${'distanciaEntradaPunto'}.${p} , 8);
172
173                                     }else{
174
... ${'punto'}.${p}.${tramo}.${tram}.${seccion}.${sec} = array
... ("x"=>(2*$radio)-$radio*(1-abs(cos($p*($radianes/$numQuesitos)+$radianes))), "y"
... =>$radio*(1-abs(sin($p*($radianes/$numQuesitos)+$radianes))));
175                                     }
176                                     //distancia entre el punto de
... entrada en la seccion y otro pto
177                                     }
178
179                                     break;
180
181                                     case 2://Trapezio
182                                     $numPuntos = 4;
183                                     $sup2 =
... $_POST['supTrapezio'].$tram.'-'. $sec];
184                                     $alt2 =
... $_POST['altTrapezio'].$tram.'-'. $sec];
185                                     $base =
... $_POST['baseTrapezio'].$tram.'-'. $sec];
186
187                                     for ($p=0; $p <4 ; $p++) {
188                                     switch ($p) {
189                                     case 0:
190
... ${'punto'}.${p}.${tramo}.${tram}.${seccion}.${sec} = array ("x"=>0, "y"=>$alt2);
191
... ${'distanciaEntradaPunto'}.${p} = 0.5*$sup2;
192                                     break;
193                                     case 1:
194
... ${'punto'}.${p}.${tramo}.${tram}.${seccion}.${sec} = array
... ("x"=>0.5*($sup2-$base), "y"=>0);
195
... ${'distanciaEntradaPunto'}.${p} = 0.5*$base;
196                                     break;
197                                     case 2:
198
... ${'punto'}.${p}.${tramo}.${tram}.${seccion}.${sec} = array ("x"=>$sup2, "y"=>$alt2);
199                                     break;
200                                     case 3:
201
... ${'punto'}.${p}.${tramo}.${tram}.${seccion}.${sec} = array
... ("x"=>0.5*($sup2-$base)+$base, "y"=>0);
202                                     break;
203                                     }
204                                     }

```

```

205
206         break;
207
208         case 3://Triángulo
209             $numPuntos = 2;
210             $sup3 =
... $_POST['supTriangulo'].$tram.'- '.$sec];
211             $alt3 =
... $_POST['altTriangulo'].$tram.'- '.$sec];
212
213             for ($p=0; $p < 3 ; $p++) {
214                 switch ($p) {
215                     case 0:
216
... ${'punto'.$p.'.'tramo'.$tram.'.'seccion'.$sec} = array ("x"=>0,"y"=>$alt3);
217
... ${'distanciaEntradaPunto'.$p} = 0.5*$sup3;
218                                     break;
219
220                                     case 1:
221
... ${'punto'.$p.'.'tramo'.$tram.'.'seccion'.$sec} = array ("x"=>($sup3 /2),"y"=>0);
222                                     break;
223
224                                     case 2:
225
... ${'punto'.$p.'.'tramo'.$tram.'.'seccion'.$sec} = array ("x"=>$sup3,"y"=>$alt3);
226                                     break;
227                                     }
228                                 }
229
230             break;
231
232             default:
233
234             echo 'no has seleccionado una forma de la
... seccion del tramo '.$tram;
235         }
236
237         $n=0;
238
239         for ($p=0; $p < $numPuntos/2; $p++) {
240
241
242             if (isset('${'distanciaEntradaPunto'.$p}') &&
... ${'distanciaEntradaPunto'.$p} != 0) {
243
244                 //puntos de entrada
245                 ${'entrada'.$tram} = array
... ("x"=>$_POST['entradaX'.$tram],"y"=>$_POST['entradaY'.$tram],"z"=>$_POST[ '
... entradaZ'.$tram]);
246                 ${'salida'.$tram} = array

```

```

246... ("x"=>$_POST['salidaX'].$tram,"y"=>$_POST['salidaY'].$tram,"z"=>$_POST['salidaZ
... '.$tram]);
247
248 //Aqui calculos de rectas
249     ${'direccionXtram'.$tram} =
... ${'salida'.$tram}["x"]-${'entrada'.$tram}["x"];
250     ${'direccionYtram'.$tram} =
... ${'salida'.$tram}["y"]-${'entrada'.$tram}["y"];
251     ${'direccionZtram'.$tram} =
... ${'salida'.$tram}["z"]-${'entrada'.$tram}["z"];
252
253     $direccionCanal = array
... ("x"=>${'direccionXtram'.$tram},"y"=>${'direccionYtram'.$tram},"z"=>${'
... direccionZtram'.$tram});
254
255
256     if (${'direccionYtram'.$tram}!=0) {
257         $direccionPerpendicular = array
... ("x"=>${'direccionYtram'.$tram}*1,"y"=>${'direccionXtram'.$tram});
258     }elseif (${'direccionXtram'.$tram}!=0)
... {
259         $direccionPerpendicular = array
... ("x"=>${'direccionYtram'.$tram},"y"=>${'direccionXtram'.$tram}*1);
260     }
261
262     //Ecuacion de la recta
263     if
... ($direccionPerpendicular["y"]==0||$direccionPerpendicular["x"]!=0) {
264         $divisionDireccionPerpendicular =
... $direccionPerpendicular["y"]/$direccionPerpendicular["x"];
265
266         $terminoInd1EcRecta =
... -1*${'entrada'.$tram}["x"]*$divisionDireccionPerpendicular;
267
268         $terminoInd2EcRecta =
... ${'entrada'.$tram}["y"];
269
270     }elseif
... ($direccionPerpendicular["x"]==0||$direccionPerpendicular["y"]!=0) {
271
272         $divisionDireccionPerpendicular =
... $direccionPerpendicular["x"]/$direccionPerpendicular["y"];
273
274         $terminoInd1EcRecta =
... -1*${'entrada'.$tram}["y"]*$divisionDireccionPerpendicular;
275
276         $terminoInd2EcRecta =
... ${'entrada'.$tram}["x"];
277
278     }
279
280     $terminoIndEcRecta =

```

```

280... $terminoInd1EcRecta+$terminoInd2EcRecta;
281
282         //cosntruimos la ec de segundo grado:
283         $cudradoDistanciaEntradaPunto =
... pow(${ 'distanciaEntradaPunto' .$p}, 2);
284
285         $a =
... 1+pow($divisionDireccionPerpendicular, 2);
286
287         if
... ($direccionPerpendicular["y"]==0 || $direccionPerpendicular["x"]!=0) {
288
289                 $interiorParentesis =
... $terminoIndEcRecta - ${ 'entrada' .$tram}["y"];
290                 $b =
... -2*${ 'entrada' .$tram}["x"]+(2*$interiorParentesis*$
... divisionDireccionPerpendicular);
291                 $c = pow(${ 'entrada' .$tram}["x"], 2) +
... pow($interiorParentesis, 2)-$cudradoDistanciaEntradaPunto;
292
293
294                 }elseif
... ($direccionPerpendicular["x"]==0 || $direccionPerpendicular["y"]!=0) {
295
296                 $interiorParentesis =
... $terminoIndEcRecta - ${ 'entrada' .$tram}["x"];
297                 $b =
... -2*${ 'entrada' .$tram}["y"]+(2*$interiorParentesis*$
... divisionDireccionPerpendicular);
298                 $c = pow(${ 'entrada' .$tram}["y"], 2) +
... pow($interiorParentesis, 2)-$cudradoDistanciaEntradaPunto;
299
300                 }
301
302         //Ecuacion de segundo grado
303         $neg = -1;
304
305         $menosb = $b * $neg;
306         $oper1 = pow($b,2);
307         $oper2 = 4*$a*$c;
308         $resta = $oper1-$oper2;
309         $raiz = pow($resta,(1/2));
310         $dosa = 2*$a;
311
312         if ($p !=0) {
313             $n=$n+1;
314         }
315
316         if
... ($direccionPerpendicular["y"]==0 || $direccionPerpendicular["x"]!=0) {
317
318             ${ 'resultx' .$n} = round(($menosb -

```

```

318... $raiz)/$dosa,4);
319         ${'resulty'.'. $n} =
... round(${'resultx'.'. $n}*$divisionDireccionPerpendicular + $terminoIndEcRecta,4);
320     }elseif
... ($direccionPerpendicular["x"]==0||$direccionPerpendicular["y"]!=0) {
321         ${'resulty'.'. $n} = round(($menosb -
... $raiz)/$dosa,4);
322         ${'resultx'.'. $n} =
... round(${'resulty'.'. $n}*$divisionDireccionPerpendicular + $terminoIndEcRecta,4);
323     }
324
325     ${'c'.'. $tram.'s'.'. $sec.'p'.'. $n} = array (
... //construyo el array de coordenadas de cada punto.
326
... "dxc".'. $tram.'s'.'. $sec.'p'.'. $n=>${'resultx'.'. $n} ,
327
... "dyc".'. $tram.'s'.'. $sec.'p'.'. $n=>${'resulty'.'. $n} ,
328
... "dzc".'. $tram.'s'.'. $sec.'p'.'. $n=>${'punto'.'. $p.'.'. $tram.'.'. $seccion'.'. $sec}["y"]+
... ${'entrada'.'. $tram}["z"]);
329
330         $n=$n+1;
331         if
... ($direccionPerpendicular["y"]==0||$direccionPerpendicular["x"]!=0) {
332
333             ${'resultx'.'. $n} = round(($menosb +
... $raiz)/$dosa,4);
334             ${'resulty'.'. $n} =
... round(${'resultx'.'. $n}*$divisionDireccionPerpendicular + $terminoIndEcRecta,4);
335
336
337             }elseif
... ($direccionPerpendicular["x"]==0||$direccionPerpendicular["y"]!=0) {
338                 ${'resulty'.'. $n} = round(($menosb +
... $raiz)/$dosa,4);
339                 ${'resultx'.'. $n} =
... round(${'resulty'.'. $n}*$divisionDireccionPerpendicular + $terminoIndEcRecta,4);
340             }
341             ${'c'.'. $tram.'s'.'. $sec.'p'.'. $n} = array
... (//construyo el array de coordenadas de cada punto.
342
... "dxc".'. $tram.'s'.'. $sec.'p'.'. $n=>${'resultx'.'. $n} ,
343
... "dyc".'. $tram.'s'.'. $sec.'p'.'. $n=>${'resulty'.'. $n} ,
344
... "dzc".'. $tram.'s'.'. $sec.'p'.'. $n=>${'punto'.'. $p.'.'. $tram.'.'. $seccion'.'. $sec}["y"]+
... ${'entrada'.'. $tram}["z"]);
345         }
346     }
347
... //////////////////////////////////////
348                                     //////////////////////////////////////EMPIEZA LA

```

```

348... ESCRITURA/////////////////////////////////
349
... ///////////////////////////////////
350
351
... $archivoSecciones=fopen($rutaArchivos."sections.input","a") or
... die("Problemas");
352
... fwrite($archivoSecciones,$_POST['forma'].$tram.'-1')." SectionType");
353                                     fwrite($archivoSecciones,"\n");
354                                     switch ($forma) {
355                                         case 1:
356
... fwrite($archivoSecciones,$_POST['radio'].$tram.'-1')." radio");
357                                         if ($tram!=$numTramos) {
358                                             fwrite($archivoSecciones,"\n");
359                                         }
360                                         break;
361
...                                     case 2:
362
... fwrite($archivoSecciones,$_POST['supTrapezio'].$tram.'-1')." supTrap");
363                                         fwrite($archivoSecciones,"\n");
364
... fwrite($archivoSecciones,$_POST['altTrapezio'].$tram.'-1')." altTrap");
365                                         fwrite($archivoSecciones,"\n");
366
... fwrite($archivoSecciones,$_POST['baseTrapezio'].$tram.'-1')." baseTrap");
367                                         if ($tram!=$numTramos) {
368                                             fwrite($archivoSecciones,"\n");
369                                         }
370                                         break;
371
...                                     case 3:
372
... fwrite($archivoSecciones,$_POST['supTriangulo'].$tram.'-1')." supTri");
373                                         fwrite($archivoSecciones,"\n");
374
... fwrite($archivoSecciones,$_POST['altTriangulo'].$tram.'-1')." altTri");
375                                         if ($tram!=$numTramos) {
376                                             fwrite($archivoSecciones,"\n");
377                                         }
378                                         break;
379                                     }
380                                     }
381
... $archivoEntradas=fopen($rutaArchivos."entries.input","a") or die("Problemas");
382
...     if (isset($_POST['entrada'].$tram)) {
383         fwrite($archivoEntradas,$tram);
384         if ($tram!=$numTramos) {
385             fwrite($archivoEntradas,"\n");
386         }
387     }
388 }

```

```

389
390
... $archivo=fopen($rutaArchivos."geometryXYZ.input","a") or die("Problemas");
391 //vamos añadiendo el contenido
392 $numPuntosArchivo = $numPuntos+1;
393 fwrite($archivo,"NCELLS ".$NCELLS);
394 fwrite($archivo,"\n");
395 fwrite($archivo,"MANNING ".$Manning);
396 fwrite($archivo,"\n");
397 fwrite($archivo,"MIN_DEPTH ".$scalado);
398 fwrite($archivo,"\n");
399 fwrite($archivo, "N_SECTIONS 2");
400 fwrite($archivo,"\n");
401 fwrite($archivo, "N_POINTS
... ".$numPuntosArchivo);
402
403
404
405
406
407
... {
408
409
410
411
412
413
... round(${'c'}.${tram}.'s'.'.${sec}.'p'.'.${j}}['dxc'}.${tram}.'s'.'.${sec}.'p'.'.${j}],4)."
... ".round(${'c'}.${tram}.'s'.'.${sec}.'p'.'.${j}}['dyc'}.${tram}.'s'.'.${sec}.'p'.'.${j}],4)."
... ".round(${'c'}.${tram}.'s'.'.${sec}.'p'.'.${j}}['dzc'}.${tram}.'s'.'.${sec}.'p'.'.${j}],4));
409
410
411
412
413
414
... round(${'entrada'}.${tram}["x"],4)." ".round(${'entrada'}.${tram}["y"],4)."
... ".round(${'entrada'}.${tram}["z"],4));
414
415
416
417
... ) {
418
419
420
421
422
423
424
425
... ".$numPuntosArchivo);
426
427

```

```

428         $sec=2;
429         for ($n=0; $n < $numPuntos; $n++) {
430             ${'c'}.${tram}.'s'.'. $sec.'p'.'. $n} =
... array(
431                 'dxc'}.${tram}.'s'.'. $sec.'p'.'. $n=>
... ${'c'}.${tram}.'s1p'.'. $n}[ 'dxc'}.${tram}.'s1p'.'. $n]+$direccionCanal["x"],
432                 'dyc'}.${tram}.'s'.'. $sec.'p'.'. $n=>
... ${'c'}.${tram}.'s1p'.'. $n}[ 'dyc'}.${tram}.'s1p'.'. $n]+$direccionCanal["y"],
433                 'dzc'}.${tram}.'s'.'. $sec.'p'.'. $n=>
... ${'c'}.${tram}.'s1p'.'. $n}[ 'dzc'}.${tram}.'s1p'.'. $n]+$direccionCanal["z"]);
434             }
435
436         for ($n=0; $n <= 2; $n++) {
437             if ($n==0) {
438                 for ($j=0; $j <$numPuntos-1 ; )
... {
439                     fwrite($archivo,
... round('${'c'}.${tram}.'s'.'. $sec.'p'.'. $j}[ 'dxc'}.${tram}.'s'.'. $sec.'p'.'. $j],4)."
... ".round('${'c'}.${tram}.'s'.'. $sec.'p'.'. $j}[ 'dyc'}.${tram}.'s'.'. $sec.'p'.'. $j],4)."
... ".round('${'c'}.${tram}.'s'.'. $sec.'p'.'. $j}[ 'dzc'}.${tram}.'s'.'. $sec.'p'.'. $j],4));
440                     fwrite($archivo, "\n");
441                     $j=$j+2;
442                 }
443             }elseif ($n==1) {
444                 fwrite($archivo,
... round('${'salida'}.${tram}["x"],4)." ".round('${'salida'}.${tram}["y"],4)."
... ".round('${'salida'}.${tram}["z"],4));
445                 fwrite($archivo, "\n");
446             }
447             elseif ($n==2) {
448                 for ($j=$numPuntos-1; $j > 0 ;
... ) {
449                     fwrite($archivo,
... round('${'c'}.${tram}.'s'.'. $sec.'p'.'. $j}[ 'dxc'}.${tram}.'s'.'. $sec.'p'.'. $j],4)."
... ".round('${'c'}.${tram}.'s'.'. $sec.'p'.'. $j}[ 'dyc'}.${tram}.'s'.'. $sec.'p'.'. $j],4)."
... ".round('${'c'}.${tram}.'s'.'. $sec.'p'.'. $j}[ 'dzc'}.${tram}.'s'.'. $sec.'p'.'. $j],4));
450                     fwrite($archivo, "\n");
451                     $j=$j-2;
452                 }
453             }
454         }
455
456         if ($tram==1) {
457             $archivoInitial=fopen($rutaArchivos."initial.input","a") or die("Problemas");
458             }
459             if ($_POST['hInicial'].$tram]>0) {
460
461             fwrite($archivoInitial,"generate constant depth ".$_POST['hInicial'].$tram);
462             }elseif
... ($_POST['hInicial'].$tram]==0) {

```

```

463
464         fwrite($archivoInitial,"generate
... dry");
465     }
466     fwrite($archivoInitial,"\n");
467
468     if ($stram==$numTramos) {
469     fclose($archivoInitial);
470     }
471
472
473     //////////////////////////////////////
474     //////////////////////////////////////
475     //////////////////////////////////////
476     //////////////////////////////////////
477     //////////////////////////////////////
478     //////////////////////////////////////
479     //////////////////////////////////////
480     //////////////////////////////////////
481     //////////////////////////////////////
482     //////////////////////////////////////
483     //////////////////////////////////////
484     //////////////////////////////////////
485     //////////////////////////////////////
486     //////////////////////////////////////
487     //////////////////////////////////////
488     //////////////////////////////////////
489     //////////////////////////////////////
490     //////////////////////////////////////
491     //////////////////////////////////////
492     //////////////////////////////////////
493     //////////////////////////////////////
494     //////////////////////////////////////
495     //////////////////////////////////////
496     //////////////////////////////////////
497     //////////////////////////////////////
498     //////////////////////////////////////
499     //////////////////////////////////////
500     //////////////////////////////////////
501     //////////////////////////////////////
502     //////////////////////////////////////

```

```

463
464         fwrite($archivoInitial,"generate
... dry");
465     }
466     fwrite($archivoInitial,"\n");
467
468     if ($stram==$numTramos) {
469     fclose($archivoInitial);
470     }
471
472
473     //////////////////////////////////////
474     //////////////////////////////////////
475     //////////////////////////////////////
476     //////////////////////////////////////
477     //////////////////////////////////////
478     //////////////////////////////////////
479     //////////////////////////////////////
480     //////////////////////////////////////
481     //////////////////////////////////////
482     //////////////////////////////////////
483     //////////////////////////////////////
484     //////////////////////////////////////
485     //////////////////////////////////////
486     //////////////////////////////////////
487     //////////////////////////////////////
488     //////////////////////////////////////
489     //////////////////////////////////////
490     //////////////////////////////////////
491     //////////////////////////////////////
492     //////////////////////////////////////
493     //////////////////////////////////////
494     //////////////////////////////////////
495     //////////////////////////////////////
496     //////////////////////////////////////
497     //////////////////////////////////////
498     //////////////////////////////////////
499     //////////////////////////////////////
500     //////////////////////////////////////
501     //////////////////////////////////////
502     //////////////////////////////////////

```

```

476         echo '
477
478         <br/>
479         <input type="number" hidden value="' . $stram . '"
... id="tram">
480
481         <div class="row">
482             <div class="col-xs-6">
483
484                 <div class="row letra-pequena">
485                     <div class="col-xs-1"></div>
486                     <select class="btn dropdown-toggle
... col-xs-4" data-toggle="dropdown" name="condicion' . $stram . '"
... id="condicion' . $stram . '" >
487
488                         <option value="1">Q(t)</option>
489                         <option value="2">h(t)</option>
490                         <option
... value="3">h+z(t)</option>
491
492                         </select>
493
494                     </div>
495
496                     <br>
497
498                     <div class="row">
499                         <div class="col-xs-1 letra"></div>
500                         <input type="button" value="Añadir
... punto" class="add btn boton-anadir" id="add' . $stram . '"
... onclick="pulsarPoner(' . $stram . ')" />
501
502                         <input type="button" value="Quitar
... punto" class="add btn boton-anadir" id="remove' . $stram . '"
... onclick="pulsarQuitar(' . $stram . '); grafica(' . $stram . ')" />
503
504                         <br/>
505                         <br/>
506                     </div>
507                 </div>
508             </div>
509         </div>

```

```

503
504         <div class="col-xs-1"></div>;
505
506
507         if (isset($_POST['entrada'].$tram)) {
508
509             echo '
510             <div id="buildyourform'.$tram.'"
... class="col-xs-8"></div>;
511
512             }
513             echo '
514
515             </div>
516             </div>;
517             if (isset($_POST['entrada'].$tram)) {
518                 ${'esEntrada'.$tram}=true;
519                 echo '
... value="' . ${'esEntrada'.$tram} . '" hidden>
520                 <div id="container'.$tram.'" style="height:
... 340px;" class="col-xs-5 letra" ></div>;
521                 }
522                 echo '
523                 <div class="col-xs-1 letra"></div>
524             </div>
525             ';
526
527         }
528         echo '
529         </div>;'
530     }
531
532
533     //////////////////////////////////////
534     //Este crea junctions.input buscando las
... coincidencias////////////////////////////////
535
536     //////////////////////////////////////
537     //Buscan coincidencias entre entrada-salida, pero aqui los
... repiten
538
539     $cuentaUniones =0;
540     $vectorPuntos = array();
541     for ($tram=1; $tram <= $numTramos ; $tram++) {
542
543         //Creo un vector solo con XY.
544         ${'puntoEntradaXY'.$tram} = ${'entrada'.$tram};
545         unset(${'puntoEntradaXY'.$tram}["z"]);
546         ${'puntoSalidaXY'.$tram} = ${'salida'.$tram};
547         unset(${'puntoSalidaXY'.$tram}["z"]);
548
549         for ($x = $tram + 1; $x <= $numTramos ; $x++) { //voy comparando

```

```

547... todos los puntos
548         ${'puntoEntradaXY'.'. $x} = ${'entrada'.'. $x};
549         unset(${'puntoEntradaXY'.'. $x}["z"]);
550         ${'puntoSalidaXY'.'. $x} = ${'salida'.'. $x};
551         unset(${'puntoSalidaXY'.'. $x}["z"]);
552
553         switch (${'puntoEntradaXY'.'. $tram}) {
554             //Aquí se ve si una una entrada coincide con la entrada
... o salida de otro tramo.
555             //Y si coincide se introducen en un array
556             case ${'puntoEntradaXY'.'. $x}:
557                 $cuentaUniones++;
558                 $separadoArray = implode(" ",
... ${'puntoEntradaXY'.'. $tram});
559                 array_push($vectorPuntos,$separadoArray);
560
561                 break;
562
563             case ${'puntoSalidaXY'.'. $x}:
564
565                 $cuentaUniones++;
566                 $separadoArray = implode(" ",
... ${'puntoEntradaXY'.'. $tram});
567                 array_push($vectorPuntos,$separadoArray);
568
569                 break;
570             //Aquí se ve si una una salida coincide con la entrada
... o salida de otro tramo.
571             //Y si coincide se introducen en un array
572         }switch (${'puntoSalidaXY'.'. $tram}) {
573
574             case ${'puntoSalidaXY'.'. $x}:
575
576                 $cuentaUniones++;
577                 $separadoArray = implode(" ",
... ${'puntoSalidaXY'.'. $tram});
578                 array_push($vectorPuntos,$separadoArray);
579
580                 break;
581
582             case ${'puntoEntradaXY'.'. $x}:
583
584                 $cuentaUniones++;
585                 $separadoArray = implode(" ",
... ${'puntoSalidaXY'.'. $tram});
586                 array_push($vectorPuntos,$separadoArray);
587
588                 break;
589         }
590     }
591 }
592 //Deja los puntos unicos, para eliminar repetidos

```

```

593     $uniones = array_unique($vectorPuntos);
594
595     //Cuenta cuantos puntos de union hay
596     $numJunctions = count($uniones);
597     $arrayTramos = array();
598
599
600     //se meten en un array
601     $junctions=array();
602     for ($i=0; $i < $cuentaUniones ; $i++) {
603         if(isset($uniones[$i])){
604             array_push($junctions, $uniones[$i]);
605         }
606     }
607
608     //Se separan los puntos y se mira qué tramos lo contienen
609     for ($n=0; $n < $numJunctions; $n++) {
610
611         ${'punto'.'. $n'} = preg_split('/ +/', $junctions[$n]);
612
613         ${'puntoCoincidente'.'. $n'}
... =array("x"=>${'punto'.'. $n'}[0], "y"=>${'punto'.'. $n'}[1]);
614             $b=0;
615
616             for ($tram=1; $tram <= $numTramos ; $tram++) {
617
618                 ${'tramoUnion'.'. $tram'} = false;
619
620                 //Se mira qué tramos son los que coinciden en ese punto y
... se introducen en un array.
621                 if (((${'puntoCoincidente'.'. $n'}["x"]) ==
... (${'entrada'.'. $tram'}["x"]) && (${'puntoCoincidente'.'. $n'}["y"]) ==
... (${'entrada'.'. $tram'}["y"])) || (((${'puntoCoincidente'.'. $n'}["x"]) ==
... (${'salida'.'. $tram'}["x"]) && (${'puntoCoincidente'.'. $n'}["y"]) ==
... (${'salida'.'. $tram'}["y"]))) {
622
623                     ${'tramoUnion'.'. $tram'} = true;
624
625                     if ($b==0) {
626                         $tramosUnion = array();
627                         $b++;
628                     }
629                     array_push($tramosUnion, $tram);
630                     array_push($arrayTramos, $tram);
631
632                     ${'tramosUnion'.'. $n'} = implode(" ", $tramosUnion);
633                 }
634             }
635
636             //Se cuenta el numero de canales de la union
637             ${'cuentaCanalesUnion'.'. $n'} = count($tramosUnion);
638
639             //Se preparan los arrays para poder escribir los

```

```

638... archivos correctamente
639         ${'coincidenciaXY'.'.n'} =
... array("x"=>${'puntoCoincidente'.'.n'}["x"],"y"=>${'puntoCoincidente'.'.n'}["y"]);
640         ${'coincidenciaXY'.'.n'} = implode(" ",
... ${'coincidenciaXY'.'.n'});
641     }
642     //Se escribe el archivo junctions.input
643     $archivoJunctions=fopen($rutaArchivos."junctions.input","a") or
... die("Problemas");
644
645     fwrite($archivoJunctions,"N_JUNCTIONS ".$numJunctions);
646     fwrite($archivoJunctions, "\n");
647     for ($i=0; $i < $numJunctions; $i++) {
648     fwrite($archivoJunctions,"POINT ".${'coincidenciaXY'.'.i'});
649     fwrite($archivoJunctions, "\n");
650     fwrite($archivoJunctions, "N_CHANNELS
... ".${'cuentaCanalesUnion'.'.i'});
651     fwrite($archivoJunctions, "\n");
652     fwrite($archivoJunctions, ${'tramosUnion'.'.i'});
653     fwrite($archivoJunctions, "\n");
654     }
655
656     //Tengo que meter todos los tramos que se unen en un array que
... guarde todos valores, comprobar
657     //para todos los valores si esta ese valor y luego que cuente
... cuantas veces está.
658     // si está menos de 2 y no es entrada, es salida.
659     $sumaUniones = array_count_values($arrayTramos);
660
661     $tramosSalida = array();
662
663     for ($q=1; $q <= $numTramos ; $q++) {
664     $cuentaTramosUnidos=count($arrayTramos);
665     if (isset($sumaUniones[$q])) {
666     if (($sumaUniones[$q]==1 && !isset($_POST['entrada'.'.q'])) &&
... $sumaUniones[$q]!=2) {
667
668         ${'esSalida'.'.q'} = true;
669         echo '
670             <input type="number" name="esSalida'.'.q.'"
... value="".'.${'esSalida'.'.q'}.'" hidden>
671             '
672         array_push($tramosSalida, $q);
673     }
674     }elseif ($numTramos==1) {
675
676         ${'esSalida'.'.q'} = true;
677         echo '
678             <input type="number" name="esSalida'.'.q.'"
... value="".'.${'esSalida'.'.q'}.'" hidden>
679             '
680         array_push($tramosSalida, $q);

```

```

681     }
682 }
683
684 if($esValido == false){
685     echo '
686     <center>
687         <div class="col-xs-2"></div>
688         <div class="col-xs-8 alert alert-danger letra-grande">DEBE
... SELECCIONAR AL MENOS UN TRAMO COMO ENTRADA</div>
689         <div class="col-xs-2"></div>
690     </center>
691     ';
692 }else{
693     echo '
694     <!-- <input class="btn boton-siguiente boton-letra"
... onclick="pasarNumPuntos()" type="submit" value="Siguiente &rarr;"/> -->
695     <br/>
696         <p class="bg-info text-center boton-siguiente
... letra">Continúe definiendo las condiciones de contorno</p>
697     </div>';
698 }
699
700     echo '
701     <div id="Salida" style="display:none;">
702         <div class="row">
703             <div class="col-xs-3 tamaño-letra alinear-derecha">Tramos
... Salida_</div>
704             <div class="col-xs-7">
705                 <ul class="pagination letra mediano">';
706                 for ($tram=1;$tram<=$numTramos;$tram++){
707
708                     if(isset({'esSalida'.$tram}) &&
... {'esSalida'.$tram}==true){
709
710                         echo '
711
712                             <li><a id="myLink" href="#"
... onclick="paginarSalida('.$tram.', '.$numTramos.');">'.$tram.</a></li>
713                             ';
714                             $ultimoTramoSalida = $tram;
715                             }
716                             }
717                         echo '
718                 </ul>
719             </div>
720         </div>';
721
722     for ($tram=1; $tram <= $numTramos ; $tram++) {
723         if (isset({'esSalida'.$tram}) && {'esSalida'.$tram}==true) {
724             if($tram==$tramosSalida[0]){
725
726                 echo '

```

```

727 <div id="tramoSalida'.$tram.'" style="display:block;">';
728 }else{
729 echo '
730 <div id="tramoSalida'.$tram.'" style="display:none;">';}
731
732 echo '
733 <div class="col-xs-6">
734 <div class="row letra-pequena">
735 <div class="col-xs-1"></div>
736
737 <select class="btn dropdown-toggle col-xs-4"
... data-toggle="dropdown" name="condicionSalida'.$tram.'"
... id="condicionSalida'.$tram.'" onchange="opcionSalida('.$tram.')" >
738
739 <option value="0">Tipo de condición</option>
740 <option value="2">h+z(t)</option>
741 <option value="1">Q(h+z)</option>
742 <option value="3">Salida libre</option>
743 <option value="5">Fr</option>
744 </select>
745
746 </div>
747 </br>
748
749 <div id="Libre'.$tram.'" style="display:none;">
750 <div class="col-xs-8 alert alert-success
... letra-grande">SALIDA '.$tram.', LIBRE</div>
751
752 </div>
753 <div id="porPuntos'.$tram.'" style="display:none;">
754 <div class="row">
755 <div class="col-xs-1 letra"></div>
756 <input type="button" value="Añadir punto"
... class="add btn boton-anadir" id="addSalida'.$tram.'"
... onclick="pulsarPonerSalida('.$tram.')" />
757 <input type="button" value="Quitar punto"
... class="add btn boton-anadir" id="removeSalida'.$tram.'"
... onclick="pulsarQuitar('.$tram.');" graficaSalida('.$tram.')" />
758 <br/>
759 <br/>
760 </div>
761 <div class="row">
762
763 <div class="col-xs-1 letra"></div>
764 <div id="buildyourform'.$tram.'"
... class="col-xs-8"></div>
765
766
767 </div>
768 </div>
769 <div id="inputFroude'.$tram.'" style="display:none;">
770 </br>

```

```

771         </br>
772     </br>
773         <div class="col-xs-2"></div>
774         <div class="input-group">
775             <span class="input-group-addon"
... id="tInicial">Numero de Froude en la salida '.$tram.'</span>
776
777                 <input class="form-control"
... id="inputFroude'.$tram.'" name="inputFroude'.$tram.'" type="number" step="any"
... value="0.8" >
778             </div>
779         </div>
780     </div>
781     <div id="container'.$tram.'" style="height: 340px;"
... class="col-xs-5 letra" ></div>;
782
783         if ($tram == $ultimoTramoSalida) {
784             echo '
785                 <input class="btn boton-siguiente boton-letra"
... onclick="pasarNumPuntos()" type="submit" value="Siguiente &rarr;"/>';
786             }else{
787                 echo '
788                     <br/>
789                     <p class="bg-info text-center boton-siguiente
... letra">Continúe definiendo el tramo '.(($tram+1)).'</p>';
790                 }
791             echo '
792
793         </div>';
794     }
795 }
796 echo '
797 </div>';
798
799     if
... (isset($_POST['topologiaEdit'])&&($_POST['topologiaEdit']==1)) {
800         echo '
801             <div id="datosEdit" hidden>
802                 <input type="number" id="ccEdit" name="ccEdit" step="any"
... value="' . $_POST['topologiaEdit'] . '" hidden>
803                 <input type="number" id="numTramosEdit" name="numTramosEdit"
... value="' . $_POST['numTramosEdit'] . '" hidden>
804                 <input type="text" id="carpetaEdit" name="carpetaEdit"
... value="' . $_POST['carpetaEdit'] . '" hidden>
805                 </div>
806                 ';
807             }
808         echo '
809 </form>';
810 ?>
811 </body>
812
813 </html>

```

## Javascript

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 function pasarNumPuntos() {
4     var numPuntosTotales = (parseInt($("#span").length)) * 0.5;
5
6     divPuntos = $("<input hidden type='number' name='numCondiciones'
... value='\" + numPuntosTotales + \"'/>");
7
8     $("#form.panel-body").append(divPuntos);
9 }
10
11 function pulsarPoner(tramo) {
12
13     if (parseInt($("#div.fieldwrapper" + tramo).length) <= 0) {
14         intId = 1;
15     } else {
16         intId = (parseInt($("#div.fieldwrapper" + tramo).length)) + 1;
17     };
18
19     fieldWrapper = $("<div class='fieldwrapper' + tramo + ' campos row
... form-inline' id='field' + tramo + '-' + intId + \"'/>");
20
21
22     tamano1 = $("<div class=' input-group letra-pequena col-xs-7'>");
23
24     condicionSeleccionada = $("##condicion" + tramo + "
... option:selected").text();
25
26     span = $("<span class='input-group-addon condicion'>" +
... condicionSeleccionada + "<sub>" + intId + "</sub></span>");
27     condicion = $("<input type='number' step='any' onchange='grafica(" + tramo + ")'
... \\" aria-describedby='calado'.$tram.'\" class='form-control \"
... value='0' name='condicion' + intId + "-" + tramo + "\" id='condicion' +
... intId + "-" + tramo + \"'/>");
28
29     tamano2 = $("<div class=' input-group letra-pequena col-xs-5'>");
30
31     span2 = $("<span class='input-group-addon '>t<sub>" + intId + "</sub>
... [s]</span>");
32     tiempo = $("<input type='number' onchange='grafica(" + tramo + ")'
... step='any' class='form-control' value='0' id='tiempo' + intId + "-" +
... tramo + "\" name='tiempo' + intId + "-" + tramo + \"'/> ");
33
34     salto = $("</br>")
35
36
37     $("##buildyourform" + tramo).append(fieldWrapper);
38
39     fieldWrapper.append(tamano2);
40
41     tamano2.append(span2);
42     tamano2.append(tiempo);
```

```

43
44     fieldWrapper.append(tamano1);
45
46     tamano1.append(span);
47     tamano1.append(condicion);
48
49
50     fieldWrapper.append(salto);
51 }
52
53 function pulsarPonerSalida(tramo) {
54
55     if (parseInt($("#div.fieldwrapper" + tramo).length) <= 0) {
56
57         intId = 1;
58     } else {
59         intId = (parseInt($("#div.fieldwrapper" + tramo).length)) + 1;
60
61     };
62
63
64     fieldWrapper = $("<div class=\"fieldwrapper\" + tramo + \" campos row
... form-inline\" id=\"field\" + tramo + \"-\" + intId + \">\");
65
66
67     tamano1 = $("<div class=\" input-group letra-pequena col-xs-7\">");
68
69     condicionSeleccionada = $("#condicionSalida" + tramo + "
... option:selected").text();
70
71     span = $("<span class=\"input-group-addon condicion\">" +
... condicionSeleccionada + "<sub>" + intId + "</sub></span>");
72     condicion = $("<input type=\"number\" step=\"any\"
... onchange=\"graficaSalida(" + tramo + ")\" aria-describedby=\"calado'.$tram.'\"
... class=\"form-control \" value=\"0\" name=\"condicion\" + intId + \"-\" + tramo +
... \"\" id=\"condicion\" + intId + \"-\" + tramo + \">\");
73
74     tamano2 = $("<div class=\" input-group letra-pequena col-xs-5\">");
75
76
77     if (parseInt($("#condicionSalida" + tramo).val()) == 1) {
78         span2 = $("<span class=\"input-group-addon \">h+z<sub>" + intId +
... "</sub> [m]</span>");
79
80     } else {
81         span2 = $("<span class=\"input-group-addon \">t<sub>" + intId + "</sub>
... [s]</span>");
82     }
83
84     tiempo = $("<input type=\"number\" onchange=\"graficaSalida(" + tramo +
... ")\" step=\"any\" class=\"form-control\" value=\"0\" id=\"tiempo\" + intId +
... \"-\" + tramo + "\" name=\"tiempo\" + intId + \"-\" + tramo + \">\");

```

```

85
86     salto = $("  
");
87
88
89     $("#buildyourform" + tramo).append(fieldWrapper);
90     fieldWrapper.append(tamano2);
91     fieldWrapper.append(tamano1);
92     fieldWrapper.append(salto);
93
94
95     tamano1.append(span);
96     tamano1.append(condicion);
97
98     tamano2.append(span2);
99     tamano2.append(tiempo);
100 }
101
102 function pulsarQuitar(tramo) {
103
104     var fieldWrapper = $("

113


```

```

136
137     serie = {
138         name: tramo,
139         data: vector,
140         lineWidth: grosor
141     };
142     var JSON = [];
143
144 };
145
146 JSON.push(serie);
147
148 switch (parseInt($("#condicion" + tramo).val())) {
149     case 1:
150         unidadesSalida = "m3/s";
151         break;
152
153     case 2:
154     case 3:
155         unidadesSalida = "m";
156         break;
157
158     default:
159         unidadesSalida = "";
160 }
161 var options = {
162     chart: {
163         style: {
164             fontFamily: 'Open Sans Condensed'
165         },
166         renderTo: 'container' + tramo,
167         type: 'line',
168         zoomType: 'x'
169     },
170
171     title: {
172         text: 'Condición de entrada del tramo ' + tramo
173     },
174     xAxis: {
175         title: {
176             text: 'Tiempo'
177         }
178     },
179     yAxis: {
180         title: {
181             text: $("#condicion" + tramo + " option:selected").text() + "
... [" + unidadesSalida + "]"
182         }
183     },
184     tooltip: {
185         formatter: function() {
186             return $("#condicion" + tramo + " option:selected").text() + '

```

```

186... (t=' + this.x + 's)' + ' = ' + (this.y).toFixed(4) + unidadesSalida;
187     }
188   },
189   legend: {
190     enabled: false,
191     layout: 'vertical',
192     align: 'center',
193     verticalAlign: 'bottom',
194     borderWidth: 0
195   },
196   series: []
197 };
198
199 options.series = JSON;
200
201 // Se crea el gráfico
202 var chart = new Highcharts.Chart(options);
203
204
205 };
206
207 function graficaSalida(tramo) {
208
209   numCampos = parseInt($("#div.fieldwrapper" + tramo).length);
210
211   for (i = 1; i <= parseInt($("#div.fieldwrapper" + tramo).length); i++) {
212
213
214     if (i == 1) {
215       vector = [];
216     };
217     condicion = parseFloat($("#input#condicion" + i + "-" + tramo).val());
218     tiempo = parseFloat($("#input#tiempo" + i + "-" + tramo).val());
219
220     puntoCondicion = [tiempo, condicion];
221     vector.push(puntoCondicion);
222
223
224     tram = 'Condición tramo ' + tramo;
225     grosor = 5;
226
227
228     serie = {
229       name: tram,
230       data: vector,
231       lineWidth: grosor
232     };
233
234     var JSON = [];
235
236   };
237

```

```

238     JSON.push(serie);
239
240     switch (parseInt($("#condicionSalida" + tramo).val())) {
241         case 1:
242             unidadesSalida = "m3/s";
243             break;
244
245         case 2:
246             unidadesSalida = "m";
247             break;
248
249         default:
250             unidadesSalida = "";
251     }
252     if (parseInt($("#condicionSalida" + tramo).val()) == 1) {
253         varX = "h+z [m]";
254
255     } else {
256         varX = "Tiempo [s]";
257     }
258     var options = {
259         chart: {
260             style: {
261                 fontFamily: 'Open Sans Condensed'
262             },
263             renderTo: 'container' + tramo,
264             type: 'line',
265             zoomType: 'x'
266         },
267         title: {
268             text: 'Condición de entrada del tramo ' + tramo
269         },
270         xAxis: {
271             title: {
272                 text: varX
273             }
274         },
275         yAxis: {
276             title: {
277                 text: $("#condicionSalida" + tramo + " option:selected").text()
278                 + " [" + unidadesSalida + "]"
279             }
280         },
281         tooltip: {
282             formatter: function() {
283                 return $("#condicionSalida" + tramo + "
284                 option:selected").text() + ' (t=' + this.x + 's)' + ' = ' + (this.y).toFixed(4)
285                 + unidadesSalida;
286             }
287         },
288         legend: {
289             enabled: false,

```

```

287         layout: 'vertical',
288         align: 'center',
289         verticalAlign: 'bottom',
290         borderWidth: 0
291     },
292     series: []
293 };
294
295 options.series = JSON;
296
297 // Se crea el gráfico
298 var chart = new Highcharts.Chart(options);
299 };
300
301
302 function condicionContEnt() {
303     $("#Entrada").show();
304     $("#Salida").hide();
305 };
306
307 function condicionContSal() {
308     $("#Salida").show();
309     $("#Entrada").hide();
310 };
311
312 function opcionSalida(tramo) {
313
314     switch (parseInt($('#condicionSalida' + tramo).val())) {
315         case 1:
316             $("div#Libre" + tramo).hide();
317             $("div#porPuntos" + tramo).show();
318             $("div#inputFroude" + tramo).hide();
319             break;
320
321         case 2:
322
323             $("div#Libre" + tramo).hide();
324             $("div#porPuntos" + tramo).show();
325             $("div#inputFroude" + tramo).hide();
326
327             break;
328
329         case 3:
330             $("div#Libre" + tramo).show();
331             $("div#porPuntos" + tramo).hide();
332             $("div#inputFroude" + tramo).hide();
333
334             break;
335
336         case 5:
337
338             $("div#Libre" + tramo).hide();

```

```

339     $("div#porPuntos" + tramo).hide();
340     $("div#inputFroude" + tramo).show();
341
342     break;
343
344     default:
345
346         $("div#Libre" + tramo).hide();
347         $("div#porPuntos" + tramo).hide();
348         $("div#inputFroude" + tramo).hide();
349     }
350 };
351
352 function editCCentrada() {
353     if ($("#ccEdit").length > 0) {
354         carpeta = $("#carpetaEdit").val();
355         tramEditEntrada = 1;
356         $.ajax({
357             url: 'cloud/' + carpeta + '/inlet.input',
358             type: "get",
359             async: false,
360             success: function(dataInlet) {
361                 var lineasInlet = dataInlet.split('\n');
362                 cuentaLineasInlet = lineasInlet.length;
363                 for (var j = 0; j < cuentaLineasInlet; j) {
364
365                     if (lineasInlet[j] == "TYPE 1" || lineasInlet[j] == "TYPE
... 2" || lineasInlet[j] == "TYPE 3") {
366
367                         switch (lineasInlet[j]) {
368                             case "TYPE 1":
369                                 $("select#condicion" +
... (tramEditEntrada)).val(1);
370                                 break;
371                             case "TYPE 2":
372                                 $("select#condicion" +
... (tramEditEntrada)).val(2);
373                                 break;
374                             case "TYPE 3":
375                                 $("select#condicion" +
... (tramEditEntrada)).val(3);
376                                 break;
377                         }
378                         lineaNumPuntosCC = lineasInlet[j + 1].split(" ");
379                         numPuntosCC = parseInt(lineaNumPuntosCC[1]);
380                         primerPunto = j + 1 + 1;
381                         ultimoPunto = j + 1 + numPuntosCC;
382                         for (var i = 1; i <= numPuntosCC; i++) {
383                             $("input#add" + (tramEditEntrada)).click();
384                         }
385                         numInput = 1;
386                         for (var punto = primerPunto; punto <= ultimoPunto;

```

```

386... punto++) {
387
388         valoresCC = lineasInlet[punto].split(" ");
389         tiempo = valoresCC[0];
390         condicion = valoresCC[1];
391         $("input#tiempo" + numInput + "-" +
... tramEditEntrada).val(tiempo);
392         $("input#condicion" + numInput + "-" +
... tramEditEntrada).val(condicion);
393         numInput++;
394     }
395
396     grafica(tramEditEntrada);
397     j = j + 2 + numPuntosCC;
398     tramEditEntrada = tramEditEntrada + 1;
399
400     } else {
401         tramEditEntrada = tramEditEntrada + 1;
402         j++;
403     }
404 }
405 }
406 }).done(editCCsalida());
407 }
408 }
409
410 function editCCsalida() {
411     if ($("#ccEdit").length > 0) {
412         carpeta = ($("#carpetaEdit").val());
413         tramEditSalida = 1;
414         $.ajax({
415             url: 'cloud/' + carpeta + '/outlet.input',
416             type: "get",
417             async: false,
418             success: function(dataOutlet) {
419                 var lineasOutlet = dataOutlet.split('\n');
420                 cuentaLineasOutlet = lineasOutlet.length;
421                 for (var m = 0; m < cuentaLineasOutlet;) {
422                     if (lineasOutlet[m] == "TYPE 3") {
423                         $("select#condicionSalida" + tramEditSalida).val(3);
424                         opcionSalida(tramEditSalida);
425                         m++;
426                         tramEditSalida++;
427                     } else if (lineasOutlet[m] == "TYPE 5") {
428                         $("select#condicionSalida" + tramEditSalida).val(5);
429                         opcionSalida(tramEditSalida);
430                         lineaValorFroude = lineasOutlet[m + 1].split(" ");
431                         valorFroude = parseFloat(lineaValorFroude[1]);
432                         $("input#inputFroude" +
... tramEditSalida).val(valorFroude);
433                         m = m + 2;
434                         tramEditSalida++;

```

```

435         } else if (lineasOutlet[m] == "TYPE 1" || lineasOutlet[m]
... == "TYPE 2") {
436             switch (lineasOutlet[m]) {
437                 case "TYPE 1":
438                     $("select#condicionSalida" +
... (tramEditSalida)).val(1);
439                     break;
440                 case "TYPE 2":
441                     $("select#condicionSalida" +
... (tramEditSalida)).val(2);
442                     break;
443             }
444             opcionSalida(tramEditSalida);
445             lineaNumPuntosCCout = lineasOutlet[m + 1].split(" ");
446             numPuntosCCout = parseInt(lineaNumPuntosCCout[1]);
447             primerPuntoOut = m + 1 + 1;
448             ultimoPuntoOut = m + 1 + numPuntosCCout;
449             for (var n = 1; n <= numPuntosCCout; n++) {
450                 $("input#addSalida" + (tramEditSalida)).click();
451             }
452             numInputOut = 0;
453             for (var puntoOut = primerPuntoOut; puntoOut <=
... ultimoPuntoOut; puntoOut++) {
454                 numInputOut++;
455                 valoresCCout = lineasOutlet[puntoOut].split(" ");
456                 tiempoOut = valoresCCout[0];
457                 condicionOut = valoresCCout[1];
458                 $("input#tiempo" + numInputOut + "-" +
... tramEditSalida).val(tiempoOut);
459                 $("input#condicion" + numInputOut + "-" +
... tramEditSalida).val(condicionOut);
460             }
461             graficaSalida(tramEditSalida);
462             m = m + 2 + numPuntosCCout;
463             tramEditSalida++;
464         } else if ((lineasOutlet[m] == "TYPE 11") ||
... (lineasOutlet[m] == "")) {
465             m++;
466             tramEditSalida++;
467         }
468     }
469 }
470 })
471 }
472 }
473 $(document).ready(function() {
474
475     editCCentrada();
476 })

```

## Parámetros

### HTML

```
1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
8     <link rel="stylesheet" href="css/bootstrap.min.css">
9     <link href="fonts/googleFonts.woff2" rel="stylesheet">
10    <link rel="stylesheet" href="css/main.css">
11    <script src="js/jquery.min.js"></script>
12    <script src="js/bootstrap.min.js"></script>
13    <script src="js/funcionesParametros.js"></script>
14    <meta charset="UTF-8">
15
16    <title>Parámetros de cálculo</title>
17 </head>
18
19 <body>
20     <div class="row">
21         <div class="col-xs-1"></div>
22         <h2 class="col-xs-4 paso-plantilla">PARÁMETROS DE SIMULACIÓN</h2>
23         <h1 class="col-xs-6
... titulo-plantilla">N&nbsp;A&nbsp;V&nbsp;I&nbsp;G&nbsp;L&nbsp;I</h1>
24         <div class="col-xs-1"></div>
25     </div>
26     <div class="row">
27         <div class="col-xs-10"></div>
28
29         <button type="button" class=" btn glyphicon glyphicon-info-sign btn-sm"
... data-toggle="modal" data-target="#infoParametros" title="Información acerca de
... estos parámetros"></button>
30
31         <div class="modal fade letra" id="infoParametros">
32             <div class="modal-dialog" role="document">
33                 <div class="modal-content">
34                     <div class="modal-header">
35                         <button type="button" class="close"
... data-dismiss="modal" aria-label="Close">
36                             <span
... aria-hidden="true">&times;</span>
37                     </button>
38                     <h4 class="modal-title" id="myModalLabel">Información
... sobre parámetros</h4>
39                 </div>
40                 <div class="modal-body">
41                     Tiempo inicial: tiempo en el que empieza la simulación.
42                     <br>
43                     Tiempo final: tiempo en el que termina la simulación.
44                     <br>
45                     CFL: número de Courant-Friedrich-Levy .
```

```

46         </br>
47         Inter. data: intervalo de tiempo entre cada dato, la
... herramienta no permite generar más de 100 instantes.
48         </br>
49         <sup>(tFinal-tInicial)</sup>&frac1;<sub>Inter.
... Data</sub>
50         <100 </br>
51
52
53     </div>
54     <div class="modal-footer">
55         <button type="button" class="btn btn-secondary"
... data-dismiss="modal">Entendido!</button>
56     </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 <?php
62
63 if (isset($_POST['topologiaEdit'])) {
64     unlink($rutaArchivos."inlet.input");
65     unlink($rutaArchivos."outlet.input");
66 }
67
68 $numTramos = $_POST['numTramos'];
69 $rutaArchivos = $_POST['rutaArchivos'];
70 $numCondiciones = $_POST['numCondiciones'];
71
72 if (is_file($rutaArchivos."inlet.input")) {
73     unlink($rutaArchivos."inlet.input");
74 }
75 }
76
77 $archivoInlet=fopen($rutaArchivos."inlet.input","a") or die("Problemas");
78
79 for ($tram=1; $tram <= $numTramos; $tram++) {
80
81
82     if (isset($_POST['esEntrada'].$tram])) {
83
84
85         ${'condicion'.$tram} = $_POST['condicion'.$tram];
86
87         switch (${'condicion'.$tram}) {
88             case 1:
89                 ${'condicion'.$tram} ="TYPE 1";
90                 fwrite($archivoInlet, ${'condicion'.$tram});
91                 fwrite($archivoInlet, "\n");
92                 $arrayCuenta = array();
93
94                 for ($i=1; $i <= $numCondiciones ; $i++) {

```

```

95         for ($t=1; $t <= $numCondiciones ; $t++) {
96             if (isset($_POST['condicion'].$t."-".$tram]) && $i==1) {
97                 array_push($arrayCuenta,
... $_POST['tiempo'].$t."-".$tram]);
98                 $numPuntosCondiciones = count($arrayCuenta);
99             }
100         }
101         if ($i==1) {
102
103             fwrite($archivoInlet, "N_POINTS ".$numPuntosCondiciones);
104             fwrite($archivoInlet, "\n");
105         }
106         if (isset($_POST['condicion'].$i."-".$tram])) {
107
108
109             fwrite($archivoInlet, $_POST['tiempo'].$i."-".$tram]);
110             fwrite($archivoInlet, " ");
111             fwrite($archivoInlet, $_POST['condicion'].$i."-".$tram]);
112             fwrite($archivoInlet, "\n");
113
114         }
115     }
116
117     break;
118
119     case 2:
120         ${'condicion'.$tram} ="TYPE 2";
121         fwrite($archivoInlet, ${'condicion'.$tram});
122         fwrite($archivoInlet, "\n");
123         $arrayCuenta = array();
124
125         for ($i=1; $i <= $numCondiciones ; $i++) {
126             for ($t=1; $t <= $numCondiciones ; $t++) {
127                 if (isset($_POST['condicion'].$t."-".$tram]) && $i==1) {
128                     array_push($arrayCuenta,
... $_POST['tiempo'].$t."-".$tram]);
129                     $numPuntosCondiciones = count($arrayCuenta);
130                 }
131             }
132             if ($i==1) {
133
134                 fwrite($archivoInlet, "N_POINTS ".$numPuntosCondiciones);
135                 fwrite($archivoInlet, "\n");
136             }
137             if (isset($_POST['condicion'].$i."-".$tram])) {
138
139
140                 fwrite($archivoInlet, $_POST['tiempo'].$i."-".$tram]);
141                 fwrite($archivoInlet, " ");
142                 fwrite($archivoInlet, $_POST['condicion'].$i."-".$tram]);
143                 fwrite($archivoInlet, "\n");
144

```

```

145     }
146   }
147
148   break;
149
150   case 3:
151     ${'condicion'}.${tram} ="TYPE 3";
152     fwrite($archivoInlet, ${'condicion'}.${tram});
153     fwrite($archivoInlet, "\n");
154     $arrayCuenta = array();
155
156     for ($i=1; $i <= $numCondiciones ; $i++) {
157       for ($t=1; $t <= $numCondiciones ; $t++) {
158         if (isset($_POST['condicion'].$t."-".$tram]) && $i==1) {
159           array_push($arrayCuenta,
... $_POST['tiempo'].$t."-".$tram]);
160           $numPuntosCondiciones = count($arrayCuenta);
161         }
162       }
163       if ($i==1) {
164
165         fwrite($archivoInlet, "N_POINTS ".$numPuntosCondiciones);
166         fwrite($archivoInlet, "\n");
167       }
168       if (isset($_POST['condicion'].$i."-".$tram])) {
169
170
171         fwrite($archivoInlet, $_POST['tiempo'].$i."-".$tram]);
172         fwrite($archivoInlet, " ");
173         fwrite($archivoInlet, $_POST['condicion'].$i."-".$tram]);
174         fwrite($archivoInlet, "\n");
175
176       }
177     }
178     break;
179
180     default:
181
182     break;
183 }//cierra switch
184
185 }else{
186   fwrite($archivoInlet, "TYPE 11");
187   fwrite($archivoInlet, "\n");
188
189 }
190 }
191
192 //////////////////////////////////////
193 //////////////////////////////////////Archivo outlet.input////////////////////////////////////
194 //////////////////////////////////////
195

```

```

196 if (is_file($rutaArchivos."outlet.input")) {
197     unlink($rutaArchivos."outlet.input");
198 }
199
200 $archivoOutlet=fopen($rutaArchivos."outlet.input","a") or die("Problemas");
201
202 for ($tram=1; $tram <= $numTramos ; $tram++) {
203
204     if (isset($_POST['esSalida'].$tram)) {
205
206         ${'condicionSalida'.$tram} = $_POST['condicionSalida'.$tram];
207
208         switch (${'condicionSalida'.$tram}) {
209
210             case 1:
211                 ${'condicionSalida'.$tram} ="TYPE 1";
212                 fwrite($archivoOutlet, ${'condicionSalida'.$tram});
213                 fwrite($archivoOutlet, "\n");
214                 $arrayCuenta = array();
215                 for ($i=1; $i <= $numCondiciones ; $i++) {
216                     for ($t=1; $t <= $numCondiciones ; $t++) {
217                         if (isset($_POST['tiempo'].$t."-".$tram]) && $i==1) {
218                             array_push($arrayCuenta,
219 ... $_POST['tiempo'].$t."-".$tram]);
220                             $numPuntosCondiciones = count($arrayCuenta);
221                         }
222                     }
223                     if ($i==1) {
224                         fwrite($archivoOutlet, "N_POINTS
225 ... ".$numPuntosCondiciones);
226                         fwrite($archivoOutlet, "\n");
227                     }
228                     if (isset($_POST['tiempo'].$i."-".$tram])) {
229
230                         fwrite($archivoOutlet, $_POST['tiempo'].$i."-".$tram]);
231                         fwrite($archivoOutlet, " ");
232                         fwrite($archivoOutlet, $_POST['condicion'].$i."-".$tram]);
233                         fwrite($archivoOutlet, "\n");
234
235                     }
236                 }
237                 break;
238
239             case 2:
240                 ${'condicionSalida'.$tram} ="TYPE 2";
241                 fwrite($archivoOutlet, ${'condicionSalida'.$tram});
242                 fwrite($archivoOutlet, "\n");
243                 $arrayCuenta = array();
244
245                 for ($i=1; $i <= $numCondiciones ; $i++) {

```

```

246         for ($t=1; $t <= $numCondiciones ; $t++) {
247             if (isset($_POST['condicion'].$t."-".$tram]) && $i==1) {
248                 array_push($arrayCuenta,
... $_POST['tiempo'].$t."-".$tram]);
249                 $numPuntosCondiciones = count($arrayCuenta);
250             }
251         }
252         if ($i==1) {
253
254             fwrite($archivoOutlet, "N_POINTS
... ".$numPuntosCondiciones);
255             fwrite($archivoOutlet, "\n");
256         }
257         if (isset($_POST['condicion'].$i."-".$tram])) {
258
259
260             fwrite($archivoOutlet, $_POST['tiempo'].$i."-".$tram]);
261             fwrite($archivoOutlet, " ");
262             fwrite($archivoOutlet, $_POST['condicion'].$i."-".$tram]);
263             fwrite($archivoOutlet, "\n");
264
265         }
266     }
267     break;
268
269     case 3:
270         ${'condicionSalida'.$tram} ="TYPE 3";
271         fwrite($archivoOutlet, ${'condicionSalida'.$tram});
272         fwrite($archivoOutlet, "\n");
273
274         break;
275
276     case 5:
277         ${'condicionSalida'.$tram} ="TYPE 5";
278         fwrite($archivoOutlet, ${'condicionSalida'.$tram});
279         fwrite($archivoOutlet, "\n");
280         fwrite($archivoOutlet, "FROUDE ".$_POST['inputFroude'].$tram]);
281         fwrite($archivoOutlet, "\n");
282
283         break;
284
285     default:
286
287         break;
288     }
289 }
290 else{
291     fwrite($archivoOutlet, "TYPE 11");
292     fwrite($archivoOutlet, "\n");
293 }
294
295 //Esta parte elimina todos los resultados anteriores

```

```

296 // (si los hay) para poder sobrescribir el caso, sólo
297 // se ejecuta en caso que vaya en modo Edición
298 if ($gestor = opendir($rutaArchivos)) {
299     if (isset($_POST['ccEdit']) && ($_POST['ccEdit'] == 1)) {
300         while (false != ($archivoBorrar = readdir($gestor))) {
301             if ($archivoBorrar != "." && $archivoBorrar != ".." &&
... $archivoBorrar != ".DS_Store") {
302                 if ($archivoBorrar != "geometryXYZ.input") {
303                     if ($archivoBorrar != "entries.input") {
304                         if ($archivoBorrar != "initial.input") {
305                             if ($archivoBorrar != "junctions.input") {
306                                 if ($archivoBorrar != "outlet.input") {
307                                     if ($archivoBorrar != "sections.input") {
308                                         if ($archivoBorrar != "inlet.input") {
309                                             if ($archivoBorrar !=
... "parameters.input") {
310
311         unlink($rutaArchivos.$archivoBorrar);
312     }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322     closedir($gestor);
323 }
324
325     echo '
326 <form action="resultado.php" method="post" enctype="multipart/form-data">
327
328
329     <input type="number" id="entradaX'.$stram.'" name="entradaX'.$stram.'"
... step="any" value="'. $_POST['entradaX'.$stram].'" hidden>
330     <input type="number" id="entradaY'.$stram.'" name="entradaY'.$stram.'"
... step="any" value="'. $_POST['entradaY'.$stram].'" hidden>
331     <input type="number" id="salidaX'.$stram.'" name="salidaX'.$stram.'"
... step="any" value="'. $_POST['salidaX'.$stram].'" hidden>
332     <input type="number" id="salidaY'.$stram.'" name="salidaY'.$stram.'"
... step="any" value="'. $_POST['salidaY'.$stram].'" hidden>
333     <input type="number" id="Ncells'.$stram.'" name="Ncells'.$stram.'" step="any"
... value="'. $_POST['Ncells'.$stram].'" hidden>';
334
335 }
336     echo '
337
338 <input name="numTramos" type="number" required="required"
... value="'. $numTramos.'" hidden/>

```

```

339 <input name="rutaArchivos" type="text" required="required"
... value="'.$rutaArchivos.'" hidden/>
340
341
342
343     <div class="row">
344         <div class="col-xs-3"></div>
345         <div class="col-xs-6">
346
347             <div id="tInicialDiv" >
348
349                 <br/>
350
351                 <div class="input-group col-xs-12">
352
353                     <span class="input-group-addon">Tiempo inicial de
... la simulación</span>
354
355                     <input class="form-control"
... aria-describedby="tInicial" type="number" name="tInicial" required="required"
... autocomplete="on" value="0" step="any" id="tInicial" min="0"/>
356                 </div>
357             </div>
358
359             <div id="tFinalDiv" >
360
361                 <br/>
362
363                 <div class="input-group col-xs-12">
364
365                     <span class="input-group-addon">Tiempo final de la
... simulación</span>
366
367                     <input class="form-control"
... aria-describedby="tFinal" type="number" name="tFinal" required="required"
... autocomplete="on" value="3600" step="any" id="tFinal" min="0"/>
368                 </div>
369             </div>
370
371             <div id="CFLDiv" >
372
373                 <br/>
374
375                 <div class="input-group col-xs-12">
376
377                     <span class="input-group-addon">CFL</span>
378
379                     <input class="form-control" aria-describedby="CFL"
... type="number" name="CFL" required="required" autocomplete="on" value="0.7"
... min="0" max="1" step="any" id="CFL" />
380                 </div>
381             </div>

```

```

382
383         <div id="interDataDiv" >
384
385             <br/>
386
387             <div class="input-group col-xs-12 interData">
388
389                 <span class="input-group-addon">Inter. Data</span>
390
391                 <input class="form-control" type="number"
... name="interData" required="required" autocomplete="on" value="100"
... id="interData" min="0"/>
392             </div>
393         </div>
394     </div>
395 </div>
396     <br/>
397     <input name="numTramos" type="number" required="required"
... value="'.$_POST['numTramos'].'" hidden/>
398     <input name="carpeta" type="text" required="required"
... value="'.$_POST['carpeta'].'" hidden/>
399
400
401     <input class="btn boton-siguiente boton-letra"
... type="submit" value="Simular &arr;" />;
402
403     if (isset($_POST['ccEdit'])&&($_POST['ccEdit'])==1) {
404         echo '
405         <div id="datosEdit" hidden>
406             <input type="number" id="parametrosEdit" name="parametrosEdit"
... step="any" value="'.$_POST['ccEdit'].'" hidden>
407             <input type="number" id="numTramosEdit" name="numTramosEdit"
... value="'.$_POST['numTramosEdit'].'" hidden>
408             <input type="text" id="carpetaEdit" name="carpetaEdit"
... value="'.$_POST['carpetaEdit'].'" hidden>
409         </div>
410         ';
411     }
412     echo '
413     </form>
414 '
415 ?>
416 </body>
417
418 </html>

```

## Javascript

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 $(document).ready(function() {
4     $("#tInicial").focus();
5
6     function warningInput() {
7         diferenciaTiempo = parseFloat($("#input#tFinal").val()) -
... parseFloat($("#input#tInicial").val());
8
9         if (parseFloat(diferenciaTiempo / parseFloat($("#interData").val())) >
... 101) {
10
11             if ($("#text-danger").length > 0) {
12                 $("#text-danger").remove();
13
14             }
15             $("#interData").attr("class", "input-group col-xs-12 interData
... has-error");
16             $("#interData").attr("title", "El número de instantes a estudiar
... demasiado alto, debe ser menor a 100");
17             $("#interDataDiv").append($("#<p class=\\"text-danger\\"
... id=\\"warningMessage\\">El número de instantes a simular demasiado alto, debe ser
... menor a 100<br> <sup>(tFinal-tInicial)</sup>&frasl;<sub>Inter.
... Data</sub><100</p>"));
18             $(".boton-siguiente").attr("disabled", "");
19         } else {
20             $(".interData").attr("class", "input-group col-xs-12 interData");
21             $(".boton-siguiente").removeAttr("disabled");
22             $(".interData").attr("title", "Intervalo de tiempo que hay entre un
... instante de tiempo y otro.");
23
24             $("#warning").remove();
25             $("#warningMessage").remove();
26         }
27     }
28
29     function editParameters() {
30         if ($("#parametrosEdit").length > 0 && $("#parametrosEdit").val() == 1)
... {
31             carpeta = $("#carpetaEdit").val();
32             $.get('cloud/' + carpeta + '/parameters.input', function(data) {
33                 var lineasParametros = data.split('\n');
34
35                 lineaTInicial = lineasParametros[1];
36                 tInicial = lineaTInicial.split(" ");
37                 tInicial = parseFloat(tInicial[1]);
38                 $("#input#tInicial").val(tInicial);
39
40                 lineaTfinal = lineasParametros[2];
41                 tFinal = lineaTfinal.split(" ");
42                 tFinal = parseFloat(tFinal[1]);
43                 $("#input#tFinal").val(tFinal);
```

```

44
45         lineaCFL = lineasParametros[3];
46         cFL = lineaCFL.split(" ");
47         cFL = parseFloat(cFL[1]);
48         $("input#cFL").val(cFL);
49
50         lineaInterData = lineasParametros[5];
51         interData = lineaInterData.split(" ");
52         interData = parseFloat(interData[1]);
53         $("input#interData").val(interData);
54     });
55 }
56 }
57
58 $("#interData").change(function() {
59     warningInput();
60 })
61 $("#input#tInicial").change(function() {
62     warningInput();
63 })
64 $("#input#tFinal").change(function() {
65     warningInput();
66 })
67 editParameters();
68
69 });

```

## Resultado

Esta página se utiliza de punto de unión entre el diseño de la red, el modelo de simulación y de los resultados.

## HTML

```
1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
8     <link rel="stylesheet" href="css/bootstrap.min.css">
9     <link href="fonts/googleFonts.woff2" rel="stylesheet">
10    <link rel="stylesheet" href="css/main.css">
11    <script src="js/jquery.min.js"></script>
12    <script src="js/bootstrap.min.js"></script>
13    <script type="text/javascript">
14    // Aquí se redirecciona a la página de resultados
15    function redireccion() {
16        $("input#siguiente").click();
17    };
18    $(document).ready(function() {
19
20        setTimeout(redireccion(), 2000);
21    });
22    </script>
23    <meta charset="UTF-8">
24
25    <title>Resultados</title>
26 </head>
27
28 <body>
29
30     <?php
31 $rutaArchivos = $_POST['rutaArchivos'];
32 $numTramos = $_POST['numTramos'];
33
34 if (is_file($rutaArchivos."parameters.input")) {
35     unlink($rutaArchivos."parameters.input");
36 }
37
38 $tInicial = $_POST['tInicial'];
39 $tFinal = $_POST['tFinal'];
40 $CFL = $_POST['CFL'];
41 $interData = $_POST['interData'];
42
43 $archivo=fopen($rutaArchivos."parameters.input","a") or die("Problemas");
44
45
46 fwrite($archivo,"N_CHAN ".$numTramos);
47 fwrite($archivo,"\n");
48 fwrite($archivo,"INITIAL_TIME ".$tInicial);
49 fwrite($archivo,"\n");
50 fwrite($archivo,"SIMULATION_TIME ".$tFinal);
51 fwrite($archivo,"\n");
```

```

52 fwrite($archivo,"CFL ".$CFL);
53 fwrite($archivo,"\n");
54 fwrite($archivo,"DUMPING_SCREEN 100");
55 fwrite($archivo,"\n");
56 fwrite($archivo,"DUMPING_DATA ".$interData);
57 fwrite($archivo,"\n");
58 fwrite($archivo,"DUMPING_QUERY ".$interData);
59 fwrite($archivo,"\n");
60 fwrite($archivo,"STEADY_TOL 1e-08");
61 fwrite($archivo,"\n");
62
63 fclose($archivo);
64
65     echo'
66 <form action="graficaPlanta.php" method="post" enctype="multipart/form-data">
67     '
68
69 for ($tram=1; $tram <= $numTramos; $tram++) {
70     echo'
71
72     <input type="number" id="entradaX'.$tram.'" name="entradaX'.$tram.'"
... step="any" value="'.$_POST['entradaX'.$tram].'" hidden>
73     <input type="number" id="entradaY'.$tram.'" name="entradaY'.$tram.'"
... step="any" value="'.$_POST['entradaY'.$tram].'" hidden>
74     <input type="number" id="salidaX'.$tram.'" name="salidaX'.$tram.'"
... step="any" value="'.$_POST['salidaX'.$tram].'" hidden>
75     <input type="number" id="salidaY'.$tram.'" name="salidaY'.$tram.'"
... step="any" value="'.$_POST['salidaY'.$tram].'" hidden>
76     <input type="number" id="Ncells'.$tram.'" name="Ncells'.$tram.'" step="any"
... value="'.$_POST['Ncells'.$tram].'" hidden>
77     '
78 }
79     echo '
80     <input type="number" id="interData" name="interData" step="any"
... value="'.$_POST['interData'].'" hidden>
81     <input type="number" id="tInicial" name="tInicial" step="any"
... value="'.$_POST['tInicial'].'" hidden>
82     <input type="number" id="tFinal" name="tFinal" step="any"
... value="'.$_POST['tFinal'].'" hidden>
83     <input name="carpeta" type="text" required="required"
... value="'.$_POST['carpeta'].'" hidden/>
84     <input name="numTramos" type="number" required="required"
... value="'.$numTramos.'" hidden/>
85     '
86 ?>
87     <div>
88         <br>
89     </div>
90     <div class="row">
91         <div class="col-xs-10"></div>
92     </div>
93     <div class="row">

```

```
94         <h1 class="col-xs-12 titulo-home">CARGANDO...</h1>
95     </div>
96     <input type="submit" value="SIGUIENTE" class="btn letra" id="siguiente"
... hidden>
97
98     </form>
99 </body>
100
101 </html>
```

## Gráficas resultado

En esta página se muestran las tres gráficas resultado explicadas en el capítulo 6 de este trabajo.

## HTML

```
1 <!DOCTYPE html>
2 <!-- Juan Carlos de Val Cremades
3 Enero 2017 -->
4 <html lang="es">
5
6 <head>
7     <link rel="shortcut icon" type="image/png" sizes="32x32"
... href="css/img/favicon-32x32.png" />
8     <link rel="stylesheet" href="css/bootstrap.min.css">
9     <link rel="stylesheet" href="css/main.css" />
10    <script src="js/jquery.min.js"></script>
11    <script src="js/bootstrap.min.js"></script>
12    <script src="js/chroma.min.js"></script>
13    <link href="fonts/googleFonts.woff2" rel="stylesheet" />
14    <script src="js/highcharts.js"></script>
15    <script src="js/exporting.js"></script>
16    <script src="js/chroma.js"></script>
17    <script src="js/multicolor_series.js"></script>
18    <script src="js/atributos.js"></script>
19    <script src="js/graficaGradiente.js"></script>
20    <meta charset="UTF-8">
21    <title>Resultados</title>
22 </head>
23
24 <body>
25
26     <div class="row">
27         <div class="col-xs-1"></div>
28         <h2 class="col-xs-4 paso-plantilla">RESULTADOS</h2>
29         <h1 class="col-xs-6
... titulo-plantilla">N&nbsp;A&nbsp;V&nbsp;I&nbsp;G&nbsp;L&nbsp;I</h1>
30         <div class="col-xs-1"></div>
31     </div>
32
33     <form action="tramos.php" method="post" enctype="multipart/form-data">
34
35         <div id="metoInputs" hidden>
36             <?php
37                 //En este if se entra en caso de venir de diseñar toda la red
... de canales
38                 //y no de cargar un caso ya generado
39                 if (isset($_POST['entradaX1'])) {
40                     $intervaloTiempo = $_POST['tFinal'] - $_POST['tInicial'];
41                     $pasoTiempo = $_POST['interData'];
42                     $tInicial = $_POST['tInicial'];
43                     $tFinal = $_POST['tFinal'];
44
45
46                     for ($tramo=1; isset($_POST['entradaX'.$tramo]); $tramo++) {
47
48                         ${'distanciaEntradaSalida'.$tramo} =
... sqrt(pow($_POST['entradaX'.$tramo]-$_POST['salidaX'.$tramo],2)+pow($_POST['
```

```

48... entradaY'.$tram]-$_POST['salidaY'.$tram],2));
49          ${'saltoCelda'.$tram} =
... ${'distanciaEntradaSalida'.$tram}/$_POST['Ncells'.$tram];
50          echo'
51          <input type="number" id="entradaX'.$tram.'" step="any"
... value="'. $_POST['entradaX'.$tram].'" hidden>
52          <input type="number" id="entradaY'.$tram.'" step="any"
... value="'. $_POST['entradaY'.$tram].'" hidden>
53          <input type="number" id="salidaX'.$tram.'" step="any"
... value="'. $_POST['salidaX'.$tram].'" hidden>
54          <input type="number" id="salidaY'.$tram.'" step="any"
... value="'. $_POST['salidaY'.$tram].'" hidden>
55          <input type="number" id="Ncells'.$tram.'"
... value="'. $_POST['Ncells'.$tram].'" hidden>
56
57          <input type="number" id="distanciaEntradaSalida'.$tram.'"
... step="any" value="'. ${'distanciaEntradaSalida'.$tram}.'" hidden>
58          <input type="number" id="saltoCelda'.$tram.'" step="any"
... value="'. ${'saltoCelda'.$tram}.'" hidden>
59          ';
60      }
61      $numTramos = $tram-1;
62
63      //Este else es en caso de venir de un caso de tique ya
... guardado con anterioridad.
64      }elseif
... (file_exists('cloud/.'$_POST['carpeta'].'/parameters.input')) {
65
66          $parametersFile =
... file('cloud/.'$_POST['carpeta'].'/parameters.input');//esto separa las lineas
... del archivo
67          $filaTramos = explode(" ", $parametersFile[0]);
68          $numTramos = trim($filaTramos[1]);
69
70          $filaTInicial = explode(" ", $parametersFile[1]);
71          $tInicial = trim($filaTInicial[1]);
72
73          $filaTfinal = explode(" ", $parametersFile[2]);
74          $tFinal = trim($filaTfinal[1]);
75
76          $filaData = explode(" ", $parametersFile[5]);
77          $dumpingData = trim($filaData[1]);
78
79
80          $intervaloTiempo = $tFinal - $tInicial;
81          $pasoTiempo = $dumpingData;
82
83
84          $file =
... file('cloud/.'$_POST['carpeta'].'/geometryXYZ.input');//esto separa las lineas
... del archivo
85

```

```

86     $siguienteTramo=0;
87     for ($i=1; $i <= $numTramos; $i++) {
88         $lineaSeccion = $siguienteTramo + 3;
89         $lineaPuntos = $siguienteTramo + 4;
90         $lineaCeldas = $siguienteTramo;
91
92         $nCellsLine = $file[$lineaCeldas];
93         $nCell = explode(" ", $nCellsLine);
94         ${'Ncells'}. $i} = trim($nCell[1]);
95
96         $nSectionsLine = $file[$lineaSeccion];
97         $nSections = explode(" ", $nSectionsLine);
98         $nSections = $nSections[1];
99
100        $nPuntosLine = $file[$lineaPuntos];
101        $nPuntos = explode(" ", $nPuntosLine);
102        $nPuntos = $nPuntos[1];
103
104        $posicionEntrada = $siguienteTramo +
... ($nPuntos*0.5)-0.5+5;
105        $entrada = explode(" ", $file[$posicionEntrada]);
106        $posicionSalida=$siguienteTramo +
... (($nPuntos*0.5)-0.5)+(6+$nPuntos);
107        $salida = explode(" ", $file[$posicionSalida]);
108
109        ${'distanciaEntradaSalida'}. $i} =
... sqrt(pow($salida[0]-$entrada[0],2) + pow($salida[1]-$entrada[1],2));
110        ${'saltoCelda'}. $i} =
... ${'distanciaEntradaSalida'}. $i}/${'Ncells'}. $i};
111
112        echo '
113
114        <input type="number" id="entradaX'$. $i.'" step="any"
... value=""$. $entrada[0].'" hidden>
115        <input type="number" id="entradaY'$. $i.'" step="any"
... value=""$. $entrada[1].'" hidden>
116        <input type="number" id="salidaX'$. $i.'" step="any"
... value=""$. $salida[0].'" hidden>
117        <input type="number" id="salidaY'$. $i.'" step="any"
... value=""$. $salida[1].'" hidden>
118        <input type="number" id="Ncells'$. $i.'" step="any"
... value=""$. ${'Ncells'}. $i}.'" hidden>
119
120        <input type="number"
... id="distanciaEntradaSalida'$. $i.'" step="any"
... value=""$. ${'distanciaEntradaSalida'}. $i}.'" hidden>
121        <input type="number" id="saltoCelda'$. $i.'"
... step="any" value=""$. ${'saltoCelda'}. $i}.'" hidden>
122        ';
123
124        $siguienteTramo = $siguienteTramo +
... $nSections*$nPuntos+6;

```

```

125         }
126     }
127 }
128
129     //Se meten todos los valores de las propiedades en un array
... para calcular maximos y minimos para
130     //fijar el gradiente de colores.
131     $arrayH=[];
132     $arrayQ=[];
133     $arrayU=[];
134     $arrayFr=[];
135     for ($tiempo=0;
... file_exists('cloud/.'.$_POST['carpeta'].'/res'.$tiempo.'_chan1.out') ;
... $tiempo++) {
136         for ($channel=1; $channel<=$numTramos ; $channel++) {
137
138             $file =
... file('cloud/.'.$_POST['carpeta'].'/res'.$tiempo.'_chan'.$channel.'.out');//esto
... separa las lineas del archivo
139             if
... (isset($_POST['Ncells'].$channel)||isset({'Ncells'.$channel})) {
140                 if (isset($_POST['Ncells'].$channel)) {
141                     $numeroCeldas=$_POST['Ncells'].$channel;
142                 }elseif (isset({'Ncells'.$channel})) {
143                     $numeroCeldas={'Ncells'.$channel};
144                 }
145                 for ($i=0; $i < $numeroCeldas; $i++) {
146
147                     $items = explode(" ", $file[$i]); //En items se
... divide cada dato en cada linea (1-9)
148                     if (!is_nan($items[1])) {
149                         array_push($arrayH, $items[1]);
150                     }
151                     if (!is_nan($items[2])) {
152                         array_push($arrayQ, $items[2]);
153                     }
154                     if (!is_nan($items[3])) {
155                         array_push($arrayU, $items[3]);
156                     }
157                     if (!is_nan($items[3])) {
158                         array_push($arrayFr, abs($items[6]));
159                     }
160                 }
161             }
162
163         }
164     }
165
166     echo'
167
168     <input type="number" id="numTramos" name="numTramos"
... value="'.$numTramos.'" hidden>

```

```

169
170         <input type="number" id="pasoTiempo" step="any"
... value="'.$pasoTiempo.'" hidden>
171         <input type="number" id="tInicial" step="any"
... value="'.$tInicial.'" hidden>
172         <input type="number" id="tFinal" step="any"
... value="'.$tFinal.'" hidden>
173
174         <input type="number" id="valueMaxH" step="any"
... value="'.max($arrayH).'" hidden>
175         <input type="number" id="valueMinH" step="any"
... value="'.min($arrayH).'" hidden>
176
177         <input type="number" id="valueMaxQ" step="any"
... value="'.max($arrayQ).'" hidden>
178         <input type="number" id="valueMinQ" step="any"
... value="'.min($arrayQ).'" hidden>
179
180         <input type="number" id="valueMaxU" step="any"
... value="'.max($arrayU).'" hidden>
181         <input type="number" id="valueMinU" step="any"
... value="'.min($arrayU).'" hidden>
182
183         <input type="number" id="valueMaxFr" step="any"
... value="'.max($arrayFr).'" hidden>
184         <input type="number" id="valueMinFr" step="any"
... value="'.min($arrayFr).'" hidden>
185
186
187         ';
188
189         echo '
190         <input name="carpeta" id="carpeta" type="text"
... required="required" value="'.$_POST['carpeta'].'" hidden/>
191         </div>
192         ';
193
194         echo '
195         <h3 class="col-xs-offset-2 letra">PLANTA_</h3>
196         <!-- todos los inputs de la grafica de planta (primera grafica) -->
197         <div class="row form-inline letra">
198             <div class=" col-xs-12">
199                 <div class="col-xs-2"></div>
200                 <div class="col-xs-3">
201                     <div class="input-group">
202                         <span class="input-group-addon">Instante de la
... simulación</span>
203                         <input class="form-control tiempos" type="number"
... id="tiempo" min="0" step="1" max="100" value="0" />
204                         <span class="input-group-addon" title="Tiempo de la
... simulación" id="tiempoSimulacionPlanta"></span>
205                     </div>

```

```

206         </div>
207
208         <p class="col-xs-3" style="text-align: right;">Propiedad</p>
209         <select name="propiedadVista" class="col-xs-4 selectpicker
... form-control" id="propiedadVista">
210             <option value="1">h</option>
211             <option value="2">Q</option>
212             <option value="3">u</option>
213             <option value="6">Fr</option>
214         </select>
215
216         <div class="col-xs-1"></div>
217     </div>
218 </div>
219
220     <!-- Línea que contiene: grafica planta, barra de escala con colores y
... valores y el botón de modificación de la escala -->
221     <div class="row lineaPlanta">
222         <div class="col-xs-1"></div>
223         <div id="containerPlanta" name="graficaPlanta" class="graficaPlanta
... col-xs-8"></div>
224         <div class="col-xs-3">
225             <div class="col-xs-2" id="botonEscalaColores">
226                 <button type="button" class="letra btn btn-sm"
... data-toggle="modal" data-target="#escalaColores" title="Escala de
... colores">Escala de colores</button>
227                 <div class="col-xs-1" id="wrap"></div>
228             </div>
229             <div class="col-xs-1"></div>
230             <div class="col-xs-1 letra" id="rangoColores"></div>
231         </div>
232     </div>
233
234     <h3 class="col-xs-offset-2 letra">TRAMO_</h3>
235     <!-- todos los inputs de la grafica de perfil (segunda grafica) -->
236     <div class="row form-inline">
237         <div class="letra col-xs-12">
238
239             <div class="col-xs-2"></div>
240             <div class="col-xs-3">
241                 <div class="input-group">
242                     <span class="input-group-addon">Instante de la
... simulación</span>
243                     <input id="tiempoPerfil" class="form-control tiempos"
... type="number" min="0" step="1" value="0" max="100" />
244                     <span class="input-group-addon" title="Tiempo de la
... simulación" id="tiempoSimulacionPerfil"></span>
245                 </div>
246             </div>
247
248             <div class="col-xs-3">
249                 <div class="input-group">

```

```

250         <span class="input-group-addon">Canal</span>
251         <input id="canalPerfil" class="form-control col-xs-3"
... type="number" min="1" value="1" max="7" />
252     </div>
253 </div>
254
255     <p class="col-xs-1" style="text-align: right;">Propiedad</p>
256     <select name="propiedadVistaPerfil" class="col-xs-3 selectpicker
... form-control" id="propiedadVistaPerfil">
257         <option value="1">h</option>
258         <option value="2">Q</option>
259         <option value="3">u</option>
260         <option value="4">A</option>
261         <option value="5">Z<sub>min</sub>+h</option>
262         <option value="6">Fr</option>
263         <option value="7">B</option>
264     </select>
265
266
267 </div>
268 </div>
269 <!-- El div en el que se coloca la gráfica de perfil (segunda grafica) -->
270 <div class="row form-inline">
271     <div class="col-xs-2"></div>
272     <div id="containerPerfil" class="chart col-xs-8"></div>
273     <div class="col-xs-2"></div>
274 </div>
275
276 <h3 class="col-xs-offset-2 letra">CELDA_</h3>
277 <!-- todos los inputs de la grafica de celda (tercera grafica) -->
278 <div class="row form-inline">
279
280     <div class="letra col-xs-12">
281
282         <div class="col-xs-2"></div>
283         <div class="col-xs-3">
284             <div class="input-group">
285                 <span class="input-group-addon">Número de celda</span>
286                 <input id="nCelda" class="form-control" type="number"
... min="1" value="1" max="4">
287                 <span class="input-group-addon" title="Distancia desde el
... principio del canal" id="distanciaCelda"></span>
288             </div>
289         </div>
290
291         <div class="col-xs-3">
292             <div class="input-group">
293                 <span class="input-group-addon">Canal</span>
294                 <input id="canalCelda" class="form-control" type="number"
... min="1" value="1" max="7">
295             </div>
296         </div>

```

```

297         <p class="col-xs-1" style="text-align: right;">Propiedad</p>
298         <select name="propiedadVistaCelda" class="col-xs-3 selectpicker
299 form-control" id="propiedadVistaCelda">
300             <option value="1">h</option>
301             <option value="2">Q</option>
302             <option value="3">u</option>
303             <option value="4">A</option>
304             <option value="5">Z<sub>min</sub>+h</option>
305             <option value="6">Fr</option>
306             <option value="7">B</option>
307         </select>
308     </div>
309 </div>
310
311 <!-- El div en el que se coloca la gráfica de celda (tercera grafica) -->
312 <div class="row">
313     <div id="containerCelda" class="chart" style="width: 840px; height:
314     400px; margin: 0 auto "></div>
315 </div>
316
317 <!-- Este es el modal en el que se puede cambiar la escala de colores a
318 nuestro gusto -->
319 <div class="modal fade letra" id="escalaColores">
320     <div class="modal-dialog">
321         <div class="modal-content">
322             <div class="modal-header">
323                 <button type="button" class="close" data-dismiss="modal"
324                 aria-label="Close">
325                     <span aria-hidden="true">&times;</span>
326                 </button>
327                 <h4 class="modal-title " id="myModallLabel">Configuración
328                 escala de colores</h4>
329             </div>
330             <div class="modal-body">
331                 Puedes cambiar el rango de valores para el gradiente de
332                 color en la gráfica que representa la planta de la red.
333                 <br>Si los valores máximo y mínimo son igual a 0, será un
334                 rango automático marcado por el mínimo y máximo valor de toda la red.
335                 <br>
336                 <br>
337                 <div class="input-group col-xs-10">
338                     <span class="input-group-addon">Mínimo valor del rango
339                     de colores</span>
340                     <input class="form-control" type="number"
341                     id="fixValueMin" step="any" value="0" />
342                 </div>
343                 <br>
344                 <div class="input-group col-xs-10">
345                     <span class="input-group-addon">Máximo valor del rango
346                     de colores</span>

```

```

339         <input class="form-control" type="number"
... id="fixValueMax" step="any" value="0" />
340     </div>
341 </div>
342     <div class="modal-footer">
343         <div class="col-xs-9"></div>
344         <button type="button" id="Aceptar" class="btn btn-secondary
... col-xs-2" data-dismiss="modal">Aceptar</button>
345     </div>
346 </div>
347 </div>
348 </div>
349     <input type="number" id="edit" name="edit" step="any" value="1" hidden>;
350
351     echo '
352     <div id="tique">
353         <p class="text-muted text-center letra">Nombre del
... tique:<br>' .$_POST['carpeta'] . '<br></p>
354         <input type="submit" value="Editar este caso" class="letra
... btn col-xs-2 col-xs-offset-5">
355         <br>
356         <br>
357         <br>
358     </div>';
359     ?>
360 </form>
361 </body>
362
363 </html>

```

## Javascript para configurar la página

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 $(document).ready(function() {
4
5     function atributosCelda() {
6         nombreCelda = '#canalCelda'.val();
7         Ncells = parseInt($('#' + nombreCelda + '').val());
8         $('#nCelda').attr("max", Ncells);
9     };
10
11    function actualizarInputs() {
12        numTramos = parseInt($('#numTramos').val());
13        tInicial = parseFloat($('#tInicial').val());
14        tFinal =
... parseFloat(($('#tFinal').val()-$('#tInicial').val())/$('#pasoTiempo').val());
15        pasoTiempo = parseFloat($('#pasoTiempo').val());
16
17        $('#canalPerfil').attr("max", numTramos);
18        $('#canalCelda').attr("max", numTramos);
19
20
21        $('#tiempo').attr("min", 0);
22        $('#tiempo').attr("max", tFinal);
23        $('#tiempo').attr("step", 1);
24        $('#tiempoPerfil').attr("min", 0);
25        $('#tiempoPerfil').attr("max", tFinal);
26        $('#tiempoPerfil').attr("step", 1);
27
28        tiempoSimulacionPlanta = tInicial + ($('#tiempo').val()*pasoTiempo);
29        $('#tiempoSimulacionPlanta').text(tiempoSimulacionPlanta.toFixed(2) + "
... s");
30        tiempoSimulacionPerfil = tInicial +
... ($('#tiempoPerfil').val()*pasoTiempo);
31        $('#tiempoSimulacionPerfil').text(tiempoSimulacionPerfil.toFixed(2) + "
... s");
32
33        atributosCelda();
34    };
35
36    actualizarInputs();
37
38    $('#canalCelda').change(function() {
39        atributosCelda();
40    });
41
42    $('#input.tiempos').change(function() {
43        actualizarInputs();
44    });
45 })
```

## Javascript encargado de la interacción con la página

```
1 // <!-- Juan Carlos de Val Cremades
2 //Enero 2017 -->
3 $(document).ready(function() {
4     $("#tiempo").focus();
5     carpeta = $("#carpeta").val();
6
7     function funcionMaxMin() {
8
9         switch (parseInt($("#propiedadVista").val())) {
10
11             case 1:
12                 defaultValueMax = parseFloat($("#valueMaxH").val());
13                 defaultValueMin = parseFloat($("#valueMinH").val());
14                 break;
15
16             case 2:
17                 defaultValueMax = parseFloat($("#valueMaxQ").val());
18                 defaultValueMin = parseFloat($("#valueMinQ").val());
19
20                 break;
21
22             case 3:
23                 defaultValueMax = parseFloat($("#valueMaxU").val());
24                 defaultValueMin = parseFloat($("#valueMinU").val());
25
26                 break;
27
28             case 6:
29                 defaultValueMax = parseFloat($("#valueMaxFr").val());
30                 defaultValueMin = parseFloat($("#valueMinFr").val());
31
32                 if ((defaultValueMax - 1) >= (1 - defaultValueMin) &&
... defaultValueMax > 1) {
33                     defaultValueMin = 1 - (defaultValueMax - 1);
34                 } else if ((defaultValueMax - 1) < (1 - defaultValueMin) &&
... defaultValueMin < 1) {
35                     defaultValueMax = 1 + (1 - defaultValueMin);
36                 }
37                 break;
38             }
39
40             if ((parseFloat($("#fixValueMin").val()) == 0) &&
... (parseFloat($("#fixValueMax").val()) == 0)) {
41                 valueMax = defaultValueMax;
42                 valueMin = defaultValueMin;
43             } else {
44                 valueMax = parseFloat($("#fixValueMax").val());
45                 valueMin = parseFloat($("#fixValueMin").val());
46             }
47         };
48
49     function funcionDatos(tiempoSeleccionado) {
```

```

50
51     propiedadVista = parseInt($("#propiedadVista").val());
52
53
54     for (var channel = 1; channel <= parseInt($("#numTramos").val());
... channel++) {
55
56         $.ajax({
57             url: 'cloud/' + carpeta + '/res' + tiempoSeleccionado + '_chan'
... + channel + '.out',
58             type: "get",
59             async: false,
60             success: function(data) {
61                 // Split the lines
62
63                 var lines = data.split('\n');
64
65                 $.each(lines, function(lineNo, line) {
66                     var items = line.split(" ");
67                     valoresRed.push(parseFloat(items[propiedadVista]));
68                 });
69
70                 if (isNaN(valoresRed[valoresRed.length - 1])) {
71                     valoresRed.splice(-1, 1);
72                 };
73             }
74         });
75
76     };
77     return valoresRed;
78 };
79
80 function funcionColor(i, valoresRed) {
81     if (parseInt($("#propiedadVista").val()) == 6) {
82         escala = chroma.scale('Spectral').domain([valueMax, valueMin]);
83     } else {
84         escala = chroma.scale(['yellow', 'navy']).mode('lch');
85     }
86     if (parseInt($("#propiedadVista").val()) == 6) {
87         color = escala(valoresRed[i]).hex();
88         diferencia = valueMax - valueMin;
89
90     } else {
91         diferencia = valueMax - valueMin;
92         numDado = valoresRed[i] - valueMin;
93         porcentaje = numDado / diferencia;
94         color = escala(porcentaje).hex();
95     }
96
97     return color;
98 };
99

```

```

100     function barraEscala(escala) {
101         $("#wrap").empty();
102         $("#rangoColores").empty();
103         for (var j = 0; j < 3; j++) {
104             $("#rangoColores").append('<br>');
105             if (j >= 1) {
106                 $("#wrap").append('<br>');
107             }
108         }
109
110         alto = ($("#graficaPlanta").height()) / 80;
111
112         if (parseInt($("#propiedadVista").val()) != 6) {
113             count = 0;
114
115             for (var i = 1; i >= 0; i = i - 2 / 100) {
116                 i = i.toFixed(3);
117                 count++;
118
119                 color = escala(i).hex();
120
121                 color = $("<div class=\"color\" + i + \"\" style=\"background:" +
... color + "; width:" + ($("#botonEscalaColores").width()) + "px; height:" + alto
... + "px\"/>");
122
123                 if (i == 1) {
124                     datoMax = $("<div id=\"maximo\" class=\"dato\" + (50 -
... count) + \"\" style=\" width:\" + ($("#botonEscalaColores").width()) + "px;
... height:\" + alto + "px\">\" + valueMax.toFixed(3) + "</div>");
125                 } else if (i == 0.5) {
126                     datoMax = $("<div id=\"medio\" class=\"dato\" + (50 -
... count) + \"\" style=\" width:\" + ($("#botonEscalaColores").width()) + "px;
... height:\" + alto + "px\">\" + ((valueMax + valueMin) / 2).toFixed(3) + "</div>");
127                 } else if (i == 0) {
128                     datoMax = $("<div id=\"minimo\" class=\"dato\" + (50 -
... count) + \"\" style=\" width:\" + ($("#botonEscalaColores").width()) + "px;
... height:\" + alto + "px\">\" + valueMin.toFixed(3) + "</div>");
129                 } else {
130                     datoMax = $("<div id=\"vacio\" class=\"dato\" + (50 -
... count) + \"\" style=\" width:\" + ($("#botonEscalaColores").width()) + "px;
... height:\" + alto + "px\">></div>");
131                 }
132
133
134                 $("#wrap").append(color);
135                 $("#rangoColores").append(datoMax);
136
137             }
138         } else {
139             count = 0;
140             for (var j = valueMax; j >= valueMin; j = j - (2 / 100) *
... diferencia) {

```

```

141         count++;
142         color = escala(j).hex();
143         color = $("<div class=\"color\" + j + \"\" style=\"background:\" +
... color + \"; width:\" + ($("#botonEscalaColores").width()) + "px; height:\" + alto
... + "px\"/>");
144
145         //Esta parte comentada sustituye que en la barra de escala de
... color salgan los valores
146         //maximo y minimo en lugar del texto Supercritico, critico y
... subcritico.
147         // if (count == 1) {
148         //     datoMax = $("<div id=\"maximo\" class=\"dato\" + (50 -
... count) + \"\" style=\" width:\" + ($("#botonEscalaColores").width()) + "px;
... height:\" + alto + "px\">" + valueMax.toFixed(5) + "</div>");
149         // } else if (count == 25) {
150         //     datoMax = $("<div id=\"medio\" style=\" width:\" +
... ($("#botonEscalaColores").width()) + "px; height:\" + alto + "px\">" +
... ((valueMax + valueMin) / 2).toFixed(5) + "</div>");
151         // } else if (count == 50) {
152         //     datoMax = $("<div id=\"minimo\" style=\" width:\" +
... ($("#botonEscalaColores").width()) + "px; height:\" + alto + "px\">" +
... valueMin.toFixed(5) + "</div>");
153         // } else {
154         //     datoMax = $("<div id=\"vacio\" style=\" width:\" +
... ($("#botonEscalaColores").width()) + "px; height:\" + alto + "px\"></div>");
155         // }
156         if (count == 1) {
157             datoMax = $("<div id=\"maximo\" class=\"dato\" + (50 -
... count) + \"\" style=\" width:\" + ($("#botonEscalaColores").width()) + "px;
... height:\" + alto + "px\">Supercritico</div>");
158         } else if (count == 25) {
159             datoMax = $("<div id=\"medio\" style=\" width:\" +
... ($("#botonEscalaColores").width()) + "px; height:\" + alto +
... "px\">Crítico</div>");
160         } else if (count == 50) {
161             datoMax = $("<div id=\"minimo\" style=\" width:\" +
... ($("#botonEscalaColores").width()) + "px; height:\" + alto +
... "px\">Subcrítico</div>");
162         } else {
163             datoMax = $("<div id=\"vacio\" style=\" width:\" +
... ($("#botonEscalaColores").width()) + "px; height:\" + alto + "px\"></div>");
164         }
165
166         $("#rangoColores").append(datoMax);
167
168         $("#wrap").append(color);
169     }
170 }
171 }
172
173 function graficaPlanta(animacion) {
174

```

```

175     numTramos = parseInt($("#numTramos").val());
176
177     funcionMaxMin();
178
179     valoresRed = [];
180
181     funcionDatos(parseFloat($("#tiempo").val()));
182     punto = 0;
183
184     switch (parseInt($("#propiedadVista").val())) {
185         case 1:
186         case 5:
187         case 7:
188             unidadesPlanta = "m";
189             break;
190
191         case 2:
192             unidadesPlanta = "m3/s";
193             break;
194
195         case 3:
196             unidadesPlanta = "m/s";
197             break;
198
199         case 4:
200             unidadesPlanta = "m2";
201             break;
202
203         default:
204             unidadesPlanta = "";
205     };
206
207     var options = {
208         chart: {
209             style: {
210                 fontFamily: 'Open Sans Condensed'
211             },
212             renderTo: 'containerPlanta',
213             defaultSeriesType: 'coloredline',
214             type: 'coloredline',
215             zoomType: 'x'
216         },
217         title: {
218             text: 'Evolución red'
219         },
220         legend: {
221             align: 'right',
222             verticalAlign: 'middle',
223             layout: 'vertical',
224             symbolPadding: 0,
225             symbolWidth: 0,
226             symbolRadius: 0

```

```

227     },
228     xAxis: {
229         title: {
230             text: 'X'
231         }
232     },
233     yAxis: {
234         title: {
235             text: 'Y'
236         }
237     },
238     function(chart) {
239         var series = chart.series;
240         $(series).each(function(i, serie) {
241             if (serie.legendSymbol) serie.legendSymbol.destroy();
242             if (serie.legendLine) serie.legendLine.destroy();
243         });
244     },
245     tooltip: {
246         crosshairs: [true, true],
247         formatter: function() {
248             return $("#propiedadVista option:selected").text() + ' (t='
... + $("#tiempo").val() + 's)' + ' = ' + this.point.myData + unidadesPlanta;
249         }
250     },
251     series: []
252 };
253
254
255     for (var tram = 1; tram <= $('#numTramos').val(); tram++) {
256
257         puntoEntrada = [parseFloat($("#input#entradaX" + tram).val()),
... parseFloat($("#input#entradaY" + tram).val())];
258
259         puntoSalida = [parseFloat($("#input#salidaX" + tram).val()),
... parseFloat($("#input#salidaY" + tram).val())];
260
261         nCells = parseInt($("#input#Ncells" + tram).val());
262
263         tramo = 'Tramo ' + tram;
264
265         grosorLinea = 6;
266
267         serie = {
268             type: 'coloredline',
269             animation: animacion,
270             name: tramo,
271             data: [],
272             lineWidth: grosorLinea,
273             marker: {
274                 enabled: false,
275                 symbol: 'circle',

```

```

276         radius: 0.5
277     }
278 };
279
280 for (var i = 0; i <= nCells; i++) {
281
282     if ((i == nCells) && (tram == $('#numTramos').val())) {
283         punto = punto - 1;
284     }
285
286     datos = new Object();
287
288     diferenciaEntradaSalida = [puntoSalida[0] - puntoEntrada[0],
... puntoSalida[1] - puntoEntrada[1]];
289
290     cadaSegmento = [(i * diferenciaEntradaSalida[0]) / nCells, (i *
... diferenciaEntradaSalida[1]) / nCells];
291
292     puntosSegmentos = [puntoEntrada[0] + cadaSegmento[0],
... puntoEntrada[1] + cadaSegmento[1]];
293
294     valorPunto = parseFloat(valoresRed[punto]);
295
296
297     datos.x = puntosSegmentos[0];
298     datos.y = puntosSegmentos[1];
299     datos.myData = valorPunto.toFixed(3);
300
301     datos.segmentColor = (funcionColor(punto, valoresRed));
302     serie.data.push(datos);
303
304     punto++;
305 }
306
307 punto = punto - 1;
308
309 options.series.push(serie);
310 }
311
312 // Create the chart
313 var chart = new Highcharts.Chart(options);
314 barraEscala(escala);
315 };
316
317
318 function datosGraficaPerfil() {
319
320     tiempoSeleccionado = parseInt($("#tiempoPerfil").val());
321
322     canalSeleccionado = parseInt($("#canalPerfil").val());
323
324     propiedadVista = parseInt($("#propiedadVistaPerfil").val());

```

```

325         minimoZ = [];
326
327         $.get('cloud/' + carpeta + '/res' + tiempoSeleccionado + '_chan' +
328         ... canalSeleccionado + '.out', function(data) {
329
330             // Se separan las líneas
331             var lines = data.split('\n');
332
333             $.each(lines, function(lineNo, line) {
334                 var items = line.split(" ");
335
336                 if (lineNo >= 0) {
337
338                     var datos1 = new Object();
339                     if (!isNaN(parseFloat(items[propiedadVista]))) {
340                         datos1.x = parseFloat(items[8]);
341                         datos1.y =
342         ... parseFloat((parseFloat(items[propiedadVista])).toFixed(4));
343                         options.series[0].data.push(datos1);
344                     }
345
346                     if (propiedadVista == 5) {
347                         var datos = new Object();
348                         if (!isNaN(items[5] - items[1])) {
349                             datos.x = parseFloat(items[8]);
350                             datos.y = parseFloat(items[5] - items[1]);
351                             options.series[1].data.push(datos);
352
353                             minimoZ.push(parseFloat(items[5] - items[1]));
354                         }
355                     }
356                 }
357             }
358
359             });
360             if (propiedadVista == 5) {
361
362                 floor = parseFloat(Math.min.apply(null, minimoZ));
363                 options.yAxis.floor = floor;
364             }
365
366             // Se crea el gráfico
367             var chart = new Highcharts.Chart(options);
368
369
370         });
371
372         switch (parseInt($("#propiedadVistaPerfil").val())) {
373             case 1:
374             case 5:

```

```

375         case 7:
376             unidadesPerfil = "m";
377             break;
378
379         case 2:
380             unidadesPerfil = "m3/s";
381             break;
382
383         case 3:
384             unidadesPerfil = "m/s";
385             break;
386
387         case 4:
388             unidadesPerfil = "m2";
389             break;
390
391         default:
392             unidadesPerfil = "";
393     };
394
395     if (propiedadVista == 5) {
396
397         var options = {
398             chart: {
399                 style: {
400                     fontFamily: 'Open Sans Condensed'
401                 },
402                 renderTo: 'containerPerfil',
403                 type: 'area',
404                 zoomType: 'x'
405             },
406             title: {
407                 text: $("#propiedadVista option:selected").text() + ' &
... longitud del canal'
408             },
409             xAxis: {
410                 title: {
411                     text: 'Longitud del canal [m]'
412                 }
413             },
414             yAxis: {
415                 title: {
416                     text: $("#propiedadVista option:selected").text() + "
... [" + unidadesPerfil + "]"
417                 }
418             },
419             tooltip: {
420                 formatter: function() {
421                     return this.series.name + ' (t=' +
... $("#tiempoPerfil").val() + 's)' + ' = ' + (this.y).toFixed(4) + unidadesPerfil
... + "; X=" + (this.x).toFixed(3) + "m";
422                 }

```

```

423     },
424     series: [{
425         name: 'Zmin+h',
426         type: 'area',
427         data: [],
428         marker: {
429             enabled: true,
430             symbol: 'circle',
431             radius: 2
432         }
433     }, {
434         name: 'Zmin',
435         type: 'area',
436         fillColor: 'rgb(191,191,191)',
437         data: [],
438         marker: {
439             enabled: true,
440             symbol: 'circle',
441             radius: 2
442         }
443     }
444 ];
445 } else {
446     var options = {
447         chart: {
448             style: {
449                 fontFamily: 'Open Sans Condensed'
450             },
451             renderTo: 'containerPerfil',
452             defaultSeriesType: 'line',
453             type: 'line',
454             zoomType: 'x'
455         },
456         title: {
457             text: $("#propiedadVistaPerfil option:selected").text() + '
... & longitud del canal'
458         },
459         legend: false,
460         xAxis: {
461             title: {
462                 text: 'Longitud del canal [m]'
463             }
464         },
465         yAxis: {
466             title: {
467                 text: $("#propiedadVistaPerfil option:selected").text()
... + " [" + unidadesPerfil + "]"
468             }
469         },
470         tooltip: {
471             formatter: function() {
472                 return $("#propiedadVistaPerfil

```

```

472... option:selected").text() + ' (t=' + $("##tiempoPerfil").val() + 's)' + ' = ' +
... (this.y).toFixed(4) + unidadesPerfil + "; X=" + (this.x).toFixed(3) + "m";
473     }
474     },
475     series: [{
476         type: 'line',
477         data: [],
478         marker: {
479             enabled: true,
480             symbol: 'circle',
481             radius: 2
482         }
483     }
484 ];
485 }
486 };
487
488
489 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
490 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
491 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
492 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
493 function datosCeldas() {
494
495     channel = parseInt($("##canalCelda").val());
496     propiedadVistaCelda = parseInt($("##propiedadVistaCelda").val());
497
498     nTiempos = parseFloat((($("##tFinal").val() - $("##tInicial").val()) /
... $("##pasoTiempo").val()) + 1);
499
500
501     celdaTiempo = [];
502     ejeTiempo = [];
503
504     for (var tiempoSeleccionado = 0; tiempoSeleccionado < nTiempos;
... tiempoSeleccionado++) {
505
506         ejeTiempo.push(parseInt(tiempoSeleccionado));
507
508         cuenta = 1;
509         $.ajax({
510             url: 'cloud/' + carpeta + '/res' + tiempoSeleccionado + '_chan'
... + channel + '.out',
511             type: "get",
512             async: false,
513             success: function(data) {
514                 // Split the lines
515                 var lines = data.split('\n');

```

```

516
517         linea = lines[parseInt($("#nCelda").val()) - 1];
518
519
520         var items = linea.split(" ");
521
522         datoDistanciaCelda = parseFloat(items[8]);
523         $("#distanciaCelda").text(datoDistanciaCelda.toFixed(3) + "
... m");
524
525         if (!isNaN(parseFloat(items[propiedadVistaCelda]))) {
526             puntoCelda = [(parseInt(tiempoSeleccionado) *
... parseFloat($("#pasoTiempo").val()),
... (parseFloat((parseFloat(items[propiedadVistaCelda])).toFixed(4)))]);
527             celdaTiempo.push(puntoCelda);
528         };
529     }
530 });
531 };
532 return celdaTiempo;
533 };
534
535
536 function graficaCelda() {
537
538     datosCeldas();
539
540     switch (parseInt($("#propiedadVistaCelda").val())) {
541         case 1:
542         case 5:
543         case 7:
544             unidadesCelda = "m";
545             break;
546
547         case 2:
548             unidadesCelda = "m3/s";
549             break;
550
551         case 3:
552             unidadesCelda = "m/s";
553             break;
554
555         case 4:
556             unidadesCelda = "m2";
557             break;
558
559         default:
560             unidadesCelda = "";
561     }
562
563     var options = {
564         chart: {

```

```

565         style: {
566             fontFamily: 'Open Sans Condensed'
567         },
568         renderTo: 'containerCelda',
569         type: 'line',
570         zoomType: 'x'
571     },
572     title: {
573         text: 'Evolución de ' + $("#propiedadVistaCelda
... option:selected").text() + ' a lo largo del tiempo de simulación'
574     },
575     legend: false,
576     xAxis: {
577         title: {
578             text: 'Tiempo de simulación [s]'
579         }
580     },
581     yAxis: {
582         title: {
583             text: $("#propiedadVistaCelda option:selected").text()
584         }
585     },
586     tooltip: {
587         formatter: function() {
588             return $("#propiedadVistaCelda option:selected").text() + '
... (t=' + this.x + 's)' + ' = ' + (this.y).toFixed(4) + unidadesCelda;
589         }
590     },
591     series: [{
592         type: 'line',
593         data: celdaTiempo,
594         marker: {
595             enabled: true,
596             symbol: 'circle',
597             radius: 2
598         }
599     }]
600 };
601
602 var chart = new Highcharts.Chart(options);
603 };
604
605
606 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
607 ////////////////////////////////////////////////////////////////////GRAFICA PLANTA/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
608 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
609
610 animacion = true;
611 graficaPlanta(animacion);
612
613
614 $("#Aceptar").click(function cambioRangoColores() {

```

```

615     animacion = true;
616     graficaPlanta(animacion);
617
618 });
619
620 $("##tiempo").change(function cambioTiempo() {
621     animacion = false;
622     graficaPlanta(animacion);
623 });
624
625 $("##propiedadVista").change(function cambioPropiedad() {
626     animacion = true;
627     graficaPlanta(animacion);
628 });
629
630
631 ///////////////////////////////////////////////////////////////////
632 ///////////////////////////////////////////////////////////////////GRAFICA PERFIL/////////////////////////////////////////////////////////////////
633 ///////////////////////////////////////////////////////////////////
634
635
636 datosGraficaPerfil();
637 ///////////////////////////////////////////////////////////////////
638 ///////////////////////////////////////////////////////////////////AQUI ES SI CAMBIA EL TIEMPO/////////////////////////////////////////////////////////////////
639 ///////////////////////////////////////////////////////////////////
640
641 $("##tiempoPerfil").change(function() {
642     datosGraficaPerfil();
643 });
644
645 ///////////////////////////////////////////////////////////////////
646 ///////////////////////////////////////////////////////////////////AQUI ES SI CAMBIA EL CANAL/////////////////////////////////////////////////////////////////
647 ///////////////////////////////////////////////////////////////////
648
649 $("##canalPerfil").change(function() {
650     datosGraficaPerfil();
651 });
652 ///////////////////////////////////////////////////////////////////
653 ///////////////////////////////////////////////////////////////////AQUI ES SI CAMBIA La PROPIEDAD/////////////////////////////////////////////////////////////////
654 ///////////////////////////////////////////////////////////////////
655 $("##propiedadVistaPerfil").change(function() {
656     datosGraficaPerfil();
657 });
658
659 ///////////////////////////////////////////////////////////////////
660 ///////////////////////////////////////////////////////////////////GRAFICA CELDA/////////////////////////////////////////////////////////////////
661 ///////////////////////////////////////////////////////////////////
662 graficaCelda();
663
664 ///////////////////////////////////////////////////////////////////
665 ///////////////////////////////////////////////////////////////////AQUI ES SI CAMBIA EL TIEMPO/////////////////////////////////////////////////////////////////
666 ///////////////////////////////////////////////////////////////////

```

```

667
668 $("#nCelda").change(function() {
669     graficaCelda();
670 });
671
672 ///////////////////////////////////////////////////////////////////
673 //AQUI ES SI CAMBIA EL CANAL////////////////////////////////////
674 ///////////////////////////////////////////////////////////////////
675
676 $("#canalCelda").change(function() {
677     graficaCelda();
678 });
679 ///////////////////////////////////////////////////////////////////
680 //AQUI ES SI CAMBIA La PROPIEDAD////////////////////////////////////
681 ///////////////////////////////////////////////////////////////////
682 $("#propiedadVistaCelda").change(function() {
683     graficaCelda();
684 });
685 });

```

## CSS para todos los documentos HTML

```
1  /*// <!-- Juan Carlos de Val Cremades
2  //Enero 2017 -->*/
3
4  html, body {
5      max-width: 100%;
6      overflow-x: hidden;
7  }
8  .quitar-float {
9      float: none;
10 }
11 .espacio-arriba {
12     margin-top: 100px;
13 }
14 .boton-home {
15     margin-top: 130px;
16     margin-left: 800px;
17     font-size: 20pt;
18     font-weight: 100;
19     font-family: 'Open Sans Condensed', sans-serif;
20 }
21 .sinPadding {
22     padding: 0;
23 }
24 .boton-letra {
25     font-size: 20pt;
26     font-weight: 100;
27     font-family: 'Open Sans Condensed', sans-serif;
28 }
29 .letra-pequena {
30     font-size: 15pt;
31     font-weight: 100;
32     font-family: 'Open Sans Condensed', sans-serif;
33 }
34 .letra-grande {
35     font-size: 40pt;
36     font-weight: 100;
37     font-family: 'Open Sans Condensed', sans-serif;
38 }
39 .boton-anadir {
40     font-size: 15pt;
41     font-weight: 100;
42     font-family: 'Open Sans Condensed', sans-serif;
43 }
44 .titulo-home {
45     margin-top: 200px;
46     margin-left: 60px;
47     font-size: 40pt;
48     font-weight: 300;
49     font-family: 'Exo', sans-serif;
50 }
51 .subtitulo-home {
52     margin-top: 5px;
```

```

53     margin-left: 60px;
54     font-size: 20pt;
55     font-weight: 100;
56     font-family: 'Open Sans Condensed', sans-serif;
57 }
58 .letra {
59     font-weight: 100;
60     font-family: 'Open Sans Condensed', sans-serif;
61 }
62 .tamano-letra {
63     font-weight: 100;
64     font-size: 25pt;
65     font-family: 'Open Sans Condensed', sans-serif;
66 }
67 .titulo-plantilla {
68     text-align: right;
69     font-size: 35pt;
70     color: #cccccc;
71     font-weight: 300;
72     font-family: 'Exo', sans-serif;
73 }
74 .paso-plantilla {
75     text-align: left;
76     font-size: 20pt;
77     font-weight: 300;
78     font-family: 'Open Sans Condensed', sans-serif;
79 }
80 .boton-siguiente {
81     position: absolute;
82     top: 87%;
83     left: 70%
84 }
85 .mediano {
86     font-size: 18px;
87 }
88 .peq-mediano {
89     font-size: 20px;
90 }
91 .alinear-izqda {
92     text-align: left;
93 }
94 .alinear-centro {
95     text-align: center;
96 }
97 .alinear-derecha {
98     text-align: right;
99 }

```