

Interreg



Cofinanciado por
la Unión Europea
Cofinanciado pela
União Europeia

RISC_PLUS

España - Portugal

Prevención, Preparación y Digitalización

**Resiliencia frente a los Riesgos de Inundaciones y Sequías
derivados del impacto del Cambio Climático en las
Cuencas Internacionales de los ríos Miño y Limia**

risc-plus.eu



Universidade de Vigo



El proyecto 0031_RISC_PLUS_6_E está cofinanciado por el Fondo Europeo de Desarrollo Regional FEDER en el marco del programa Interreg VI A España - Portugal (POCTEP) 2021-2027

INDICE DE CONTENIDOS

Informe explicativo sobre el funcionamiento de los sistemas de alerta temprana y su modelización asociada.

Entregable UVIGO: Actividad 3 . Acción 3.2..... 3

Informe explicativo sobre el funcionamiento de los sistemas de alerta temprana y su modelización asociada

1. Contexto y finalidad del sistema MIDAS

MIDAS (Miño river flood Alert System) es un sistema integrado de alerta temprana concebido para apoyar la predicción, evaluación y gestión operativa de episodios de inundación en la demarcación hidrográfica del Miño-Sil. Su desarrollo responde a la necesidad de disponer de una herramienta capaz de transformar, en un intervalo de tiempo reducido, la información meteorológica e hidrológica disponible en productos directamente utilizables para la toma de decisiones.

La cuenca del Miño-Sil incluye áreas urbanas y tramos fluviales que han sufrido históricamente inundaciones relevantes. A este condicionante se añade la previsión de una intensificación de los episodios hidrometeorológicos extremos, lo que incrementa la necesidad de contar con sistemas automáticos, reproducibles y actualizables. En este contexto, MIDAS integra la predicción hidrológica a escala de cuenca con la simulación hidráulica detallada en zonas expuestas, permitiendo anticipar no solo los caudales esperados, sino también su traducción espacial en calados, velocidades y niveles de peligrosidad.

El objetivo operativo del sistema es generar información con antelación suficiente para facilitar la adopción de medidas preventivas y de mitigación. Para ello, MIDAS combina datos observados y pronosticados, modelos numéricos y herramientas de automatización en una cadena de procesamiento común. Esta concepción integrada evita que cada modelo funcione como una aplicación aislada y permite que los resultados de una fase constituyan de forma directa las entradas de la siguiente.

2. Arquitectura funcional y flujo operativo

La arquitectura de MIDAS se organiza en torno a dos módulos de cálculo principales — hidrológico e hidráulico— conectados mediante un flujo automatizado de datos. El proceso comienza cuando están disponibles las observaciones y previsiones de precipitación, junto con la información de caudales registrada en la red de control. A partir de estas entradas, el sistema estima la evolución hidrológica de las subcuencas y calcula los caudales previsibles en los puntos y zonas de interés.

2.1. Módulo hidrológico

El módulo hidrológico utiliza HEC-HMS para representar la respuesta de la cuenca ante la precipitación y predecir la evolución de los caudales. La modelización se aplica al conjunto de la demarcación y genera series temporales en los puntos necesarios para el seguimiento hidrológico y para la alimentación del módulo hidráulico.

La información de entrada incluye precipitación observada, predicción de precipitación e información de caudales. El sistema actualiza las ventanas temporales de simulación, incorpora los datos en los archivos de cálculo y ejecuta los modelos de forma automatizada. Las series resultantes permiten disponer de una previsión de caudales con un horizonte suficiente para anticipar posibles situaciones de riesgo.

2.2. Evaluación de umbrales

Los caudales calculados en las zonas de riesgo se comparan con valores límite previamente definidos. Esta comprobación actúa como mecanismo de selección: cuando no se superan los umbrales, el proceso puede finalizar sin necesidad de lanzar cálculos hidráulicos adicionales; cuando se detecta una posible situación de riesgo, se activa el módulo hidráulico.

Esta lógica permite concentrar los recursos de cálculo en los episodios potencialmente relevantes, reduciendo tiempos de proceso y favoreciendo una respuesta operativa más eficiente.

2.3. Módulo hidráulico

El módulo hidráulico se basa en Iber+, un modelo bidimensional que resuelve la propagación del flujo en zonas urbanas y tramos fluviales susceptibles de sufrir daños por inundación. Los hidrogramas generados por HEC-HMS se incorporan como condiciones de contorno de entrada, de forma que la simulación hidráulica representa la evolución espacial del episodio previsto.

Los resultados se generan en forma de rásteres y productos cartográficos de calado, velocidad y peligrosidad. Estos productos permiten identificar áreas afectadas, zonas con mayor intensidad del flujo y elementos potencialmente expuestos, aportando una información más directamente interpretable que una serie de caudal aislada.

3. Automatización e integración de los procesos de modelización

Uno de los elementos esenciales de MIDAS es la automatización de la cadena de cálculo. La ejecución manual de modelos hidrológicos e hidráulicos exige modificar archivos, actualizar periodos de simulación, incorporar series temporales, lanzar aplicaciones y comprobar resultados. En un entorno de alerta temprana, estas operaciones deben realizarse con rapidez y con un nivel de fiabilidad que no dependa de la intervención continua de un operador.

La solución implementada utiliza scripts en Python para gestionar los archivos de configuración, incorporar la precipitación, ejecutar HEC-HMS en modo no interactivo, recuperar los hidrogramas calculados y transferirlos al modelo Iber. En el caso hidrológico, la integración con archivos HEC-DSS permite escribir las series de precipitación y leer los caudales simulados. En el caso hidráulico, se separa la información estática del modelo de los datos dinámicos de cada episodio, de modo que un mismo caso pueda reutilizarse con nuevos hidrogramas sin editar manualmente bloques extensos de condiciones de contorno.

La automatización incluye comprobaciones básicas de integridad, como la existencia de las columnas requeridas, la correcta interpretación de las fechas, el carácter numérico de los caudales y la presencia de los archivos necesarios para construir y ejecutar cada caso. Estas verificaciones contribuyen a detectar errores antes de iniciar simulaciones costosas y mejoran la trazabilidad del proceso.

El resultado es una cadena reproducible y modular. Cada componente puede ser actualizado o sustituido sin alterar el conjunto de la arquitectura, siempre que se mantengan interfaces de intercambio bien definidas. Esta característica facilita la incorporación de nuevos modelos, fuentes de datos y zonas de estudio, así como la transferencia de la metodología a otros organismos o demarcaciones.

4. Mejoras introducidas y evolución en el marco de RISC_PLUS

RISC_PLUS plantea una evolución del sistema previo orientada a ampliar sus capacidades de predicción, mejorar la robustez de los cálculos y aumentar la utilidad de los resultados para los gestores. Las líneas de trabajo se centran tanto en los modelos como en la cobertura territorial, la adquisición de información y el acceso a los productos generados.

- **Nuevos enfoques hidrológicos.** La incorporación de formulaciones más completas como el Soil Moisture Account (SMA) permite representar de manera más realista procesos como la infiltración, la evapotranspiración, la percolación y la evolución de la humedad del suelo.
- **Predicción mediante conjuntos.** El uso de diferentes previsiones de precipitación permite considerar la incertidumbre meteorológica y analizar varios escenarios posibles, en lugar de depender de una única trayectoria determinista.
- **Integración de fenómenos a gran escala.** La consideración de ríos atmosféricos aporta información adicional sobre episodios capaces de producir precipitaciones intensas y persistentes en la fachada atlántica.
- **Aplicación de inteligencia artificial.** Las técnicas de aprendizaje automático se plantean como apoyo para la predicción de aportaciones a embalses y para complementar los enfoques de modelización convencionales.
- **Ampliación de las áreas de aplicación.** La extensión de los módulos hidrológico e hidráulico permite incorporar nuevas subcuencas, tramos y núcleos expuestos, aumentando la cobertura territorial del sistema.
- **Mejora del acceso a la información.** La evolución de la interfaz y de los mecanismos de publicación busca facilitar la consulta de series, mapas y niveles de riesgo por parte de los usuarios responsables de la gestión.

5. Beneficios operativos para los organismos de cuenca

La principal aportación operativa de MIDAS es la conversión de información meteorológica e hidrológica en productos de apoyo a la decisión dentro de una única cadena de procesamiento. El sistema reduce el tiempo necesario para preparar y ejecutar los modelos, limita la edición manual de archivos y permite repetir simulaciones de forma homogénea ante sucesivas actualizaciones de las previsiones.

Para los organismos de cuenca, esta integración mejora la capacidad de seguimiento de los episodios y facilita la priorización de zonas que requieren una atención específica. Las series de caudal ayudan a evaluar la evolución prevista en la red fluvial, mientras que los mapas de calado, velocidad y peligrosidad permiten interpretar espacialmente las consecuencias potenciales y comunicar la situación a otros servicios responsables.

La disponibilidad de los resultados en un entorno web favorece el acceso compartido a la información y reduce la dependencia de aplicaciones especializadas para una primera consulta. Asimismo, la utilización de modelos y herramientas de libre acceso, junto con una estructura de cálculo basada en pocos parámetros y procesos automatizados, contribuye a contener los costes de implantación y facilita la replicabilidad.

Desde el punto de vista de la transferencia, el sistema ofrece un procedimiento documentado para conectar datos, modelos y productos cartográficos. Esta metodología puede ser adaptada a nuevos casos de estudio, siempre que se disponga de la información básica necesaria para configurar los modelos y establecer los umbrales de riesgo. De este modo, el desarrollo no se

limita a una solución cerrada, sino que constituye una base tecnológica extensible para la gestión preventiva de inundaciones.

6. Conclusión

MIDAS constituye una plataforma integrada que conecta la predicción hidrológica, la modelización hidráulica y la generación de información para la gestión del riesgo de inundación. La automatización de los procesos HEC-HMS–Iber permite pasar de los datos de precipitación y caudal a productos cartográficos y niveles de riesgo con una intervención manual reducida y dentro de plazos compatibles con un sistema de alerta temprana. Las mejoras abordadas en RISC_PLUS refuerzan esta arquitectura mediante nuevos enfoques de modelización, una mayor consideración de la incertidumbre, la ampliación territorial y una mejor accesibilidad de los resultados. En conjunto, estas actuaciones incrementan la utilidad operativa del sistema, favorecen su transferencia y consolidan una base tecnológica para apoyar la toma de decisiones ante episodios de inundación.

7. Desarrollo práctico de la cadena de modelización

Una vez descritos el contexto, la arquitectura y la utilidad operativa del sistema MIDAS, la siguiente sección presenta con mayor detalle los procedimientos técnicos empleados para automatizar la cadena de modelización hidrológica e hidráulica. Su objetivo es mostrar de forma práctica cómo se configuran, ejecutan y conectan los distintos componentes del sistema, desde la incorporación de los datos de precipitación hasta la generación de resultados hidráulicos susceptibles de ser analizados en un entorno SIG.

Informe explicativo sobre el funcionamiento de los sistemas de alerta temprana y su modelización asociada



Interreg



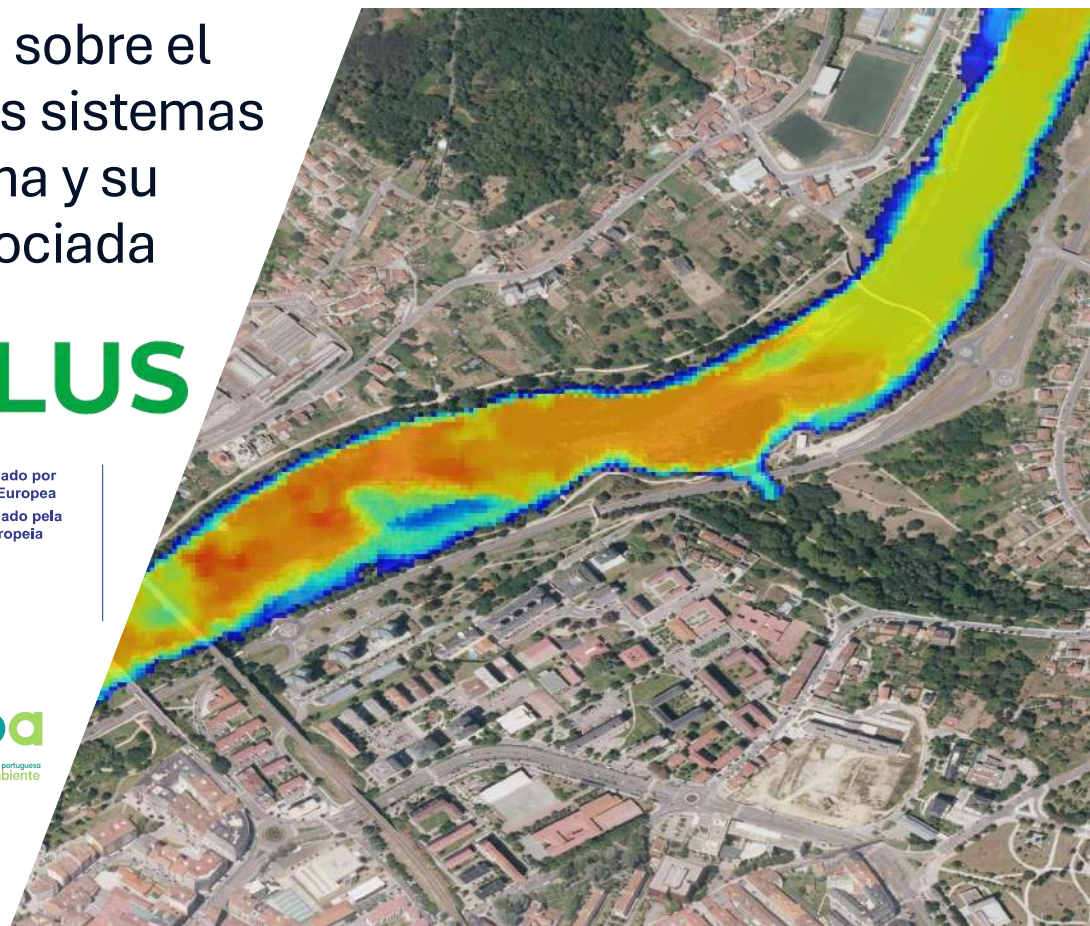
Cofinanciado por
la Unión Europea
Cofinanciado pela
União Europeia

España – Portugal

U. PORTO
FEUP FACULTY OF ENGINEERING
UNIVERSITY OF PORTO



Universidade de Vigo



Outline

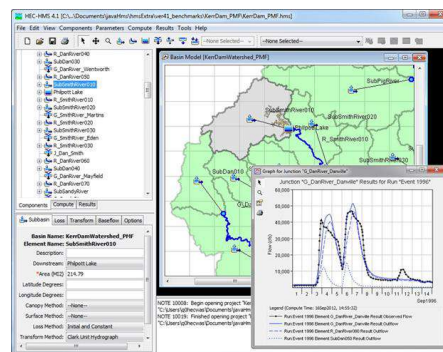
- Intro and scope
- Spyder basics and debugging
- HMS case anatomy
- Python setup and script structure
- Live execution and output inspection
- Wrap-up and Q&A

Scope and Non-Scope

- Covered in this seminar:
 - HEC-HMS file structure.
 - Minimal automation.
 - DSS input/output handling.
- Not covered in this seminar:
 - Calibration.
 - Optimization.
 - Forecast retrieval pipelines.
 - Baseflow tuning

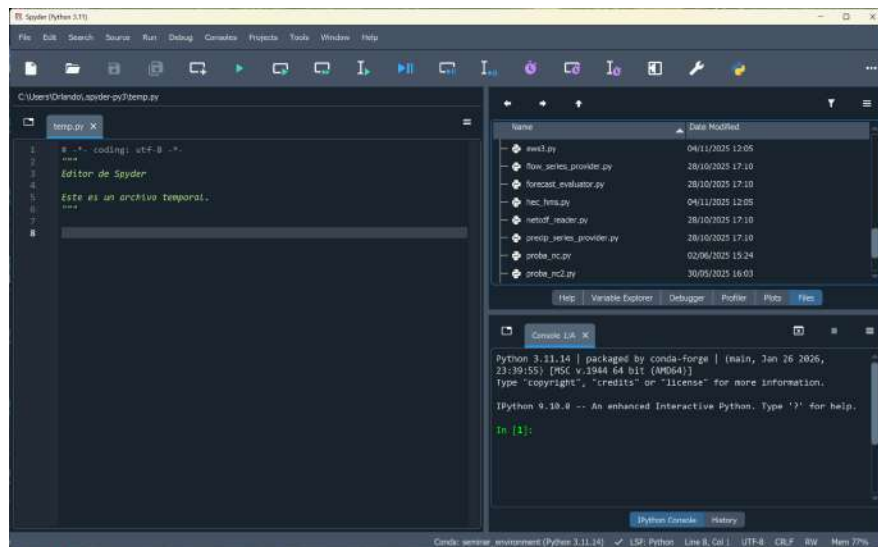
Prerequisites

- Anaconda (with seminar_environment.yml installed)
- Spyder
- HEC-HMS installed on Windows
- Uncompressed session file: seminar_auto_hms.zip



Spyder for Beginners: Interface Tour

- Left → **Editor**: where you write and modify the script
- Bottom right → **IPython Console**: where code executes and prints output
- Top right:
 - **Variable Explorer**: inspect dataframes, arrays, and scalars
 - **Plots pane**: quick visualization output
 - **File Explorer**: navigate project files
- **Reset layout**: View → Window layouts → Spyder Default Layout



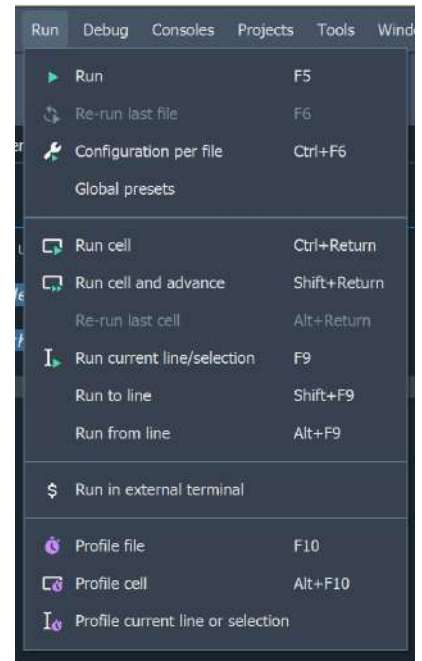
Spyder for Beginners: First Run Checklist

1. Launch Spyder from Anaconda environment
2. Check the environment in the Spyder interface



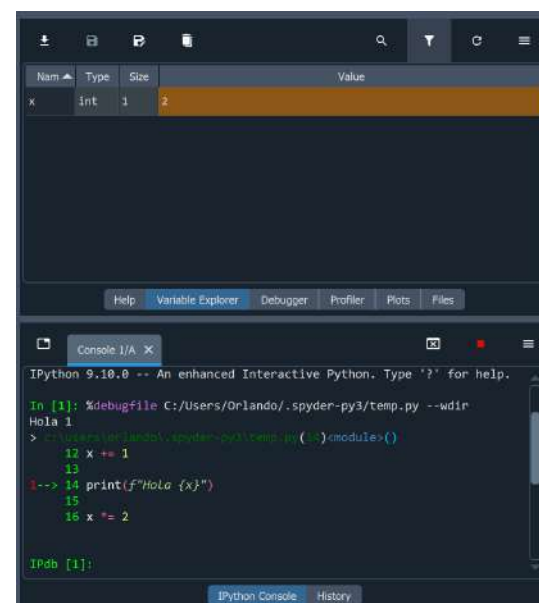
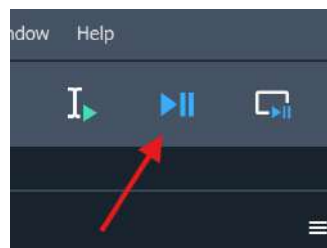
Spyder for Beginners: Ways to Execute Code

- Run full script: `F5` (Run File)
- Run selected lines: `F9`
- Run one cell (if using `# %%`): `Ctrl+Enter`



Spyder Debugging Basics

- Add breakpoint by clicking left margin in editor
- Start debugger: Debug file (bug icon or menu)
- Execution stops at breakpoints
- Inspect variables in Variable Explorer and console



Spyder Debug Controls (Essential)

- Continue: run until next breakpoint
- Step over: execute current line, stay in same function
- Step into: enter called function
- Step out: finish current function and return
- Stop: end debug session



Project Files Used Today

- Python script
 - `hms_standalone.py`
- Input rainfall
 - `precipitation.csv`
- HEC-HMS case made in the hydrological modelling session. We will **copy our HEC-HMS case into the same folder as** `hms_standalone.py`
 - `Curso_Modelizacion/`

HMS Case Files

- `.hms` project definition
- `.run` run configuration and output DSS
- `.control` simulation window and time interval
- `.gage` precipitation source and DSS pathname
- `.dss` binary input/output time-series storage
- `.basin` and `.met` basin and meteorological model parameters

Python Library Focus: `pydsstools`

- `pydsstools` is the Python bridge to **HEC-DSS files**.
- Open-source: <https://github.com/gyanz/pydsstools>
- It lets us **write rainfall** into DSS and **read simulated flow** from DSS.
- DSS **pathnames** are structured keys (A/B/C/D/E/F parts).
- In this script we are going to use:
 - `HecDss.Open`
 - `TimeSeriesContainer`
 - `put_ts`
 - `read_ts`
 - `getPathnameList`

What .hms Gives Us

- Project name
- Input DSS file name
- Basin file name
- Control file name

Code reference: `parse_hms_file`

```
Project: Curso_Modelizacion
Description:
Version: 4.12
Filepath Separator: \
DSS File Name: Curso_Modelizacion.dss
Time Zone ID: Europe/Madrid
End:

Precipitation: Meteorological_Model
Filename: Meteorological_Model.met
Description:
Last Modified Date: 2 March 2026
Last Modified Time: 19:44
End:

Basin: Sarria
Filename: Sarria.basin
Description:
Last Modified Date: 2 March 2026
Last Modified Time: 19:45
End:

Control: Control_Specifications
FileName: Control_Specifications.control
Description:
End:
```

What .run Gives Us

- Output DSS file linked to selected run name

Code reference:

`parse_run_output_dss`

```
Run: Run 1
Default Description: Yes
Log File: Run_1.log
DSS File: Run_1.dss
Is Save Spatial Results: No
Last Modified Date: 2 March 2026
Last Modified Time: 19:40:07
Last Execution Date: 3 March 2026
Last Execution Time: 08:22:29
Basin: Sarria
Precip: Meteorological_Model
Control: Control_Specifications
Time-Series Output: Save All
Time Series Results Manager Start:
Time Series Results Manager End:
End:
```

What `.gage` Gives Us

- Precipitation DSS pathname used as model input
- Single-gage setup in this seminar case

Code reference:

```
parse_precip_pathname
```

```
Gage Manager: Curso_Modelizacion
Version: 4.12
Filepath Separator: \
End:

Gage: Gage 1
  Last Modified Date: 3 March 2026
  Last Modified Time: 08:00:38
  Reference Height Units: Meters
  Reference Height: 100
  Gage Type: Precipitation
  Precipitation Type: Incremental
  Units: MM
  Data Type: PER-CUM
  Data Source Type: Manual Entry
  Variant: Variant-1
    Last Variant Modified Date: 2 March 2026
    Last Variant Modified Time: 19:39:03
    Default Variant: Yes
    DSS File Name: Curso_Modelizacion.dss
    DSS Pathname: //Gage 1/PRECIP-INC//1HOUR/GAGE/
    Start Time: 15 January 2013, 02:00
    End Time: 23 January 2013, 01:00
  End Variant: Variant-1
End:
```

What `.control` Gives Us

- The **start** and **ending datetimes** of the simulation.
- The **time step** used (in minutes): hourly in this case.

Code reference:

```
update_control_times
```

```
Control: Control_Specifications
  Last Modified Date: 2 March 2026
  Last Modified Time: 19:37:42
  Version: 4.12
  Time Zone ID: Europe/Madrid
  Time Zone GMT Offset: 3600000
  Start Date: 15 January 2013
  Start Time: 02:00
  End Date: 23 January 2013
  End Time: 01:00
  Time Interval: 60
End:
```

Configuration Block in Script

```
23 # Seminar settings: edit these values before running the script.
24 # Use project root as base so paths stay simple for students.
25
26 WINDOWS_HMS_DIR = Path(r"C:\Program Files\HEC\HEC-HMS\4.12")
27
28 PROJECT_ROOT = Path(__file__).resolve().parent
29 CASE_DIR = PROJECT_ROOT / "Curso_Modelizacion"
30 PRECIP_CSV = PROJECT_ROOT / "precipitation.csv"
31 OUT_CSV = PROJECT_ROOT / "hms_results_minimal.csv"
32
33 RUN_NAME = "Run 1"
34 TARGET_HMS_ELEMENT = "End_Point"
35 START.UTC = dt.datetime(2013, 1, 15, 1, 0, tzinfo=dt.timezone.utc)
36 SIMULATION_HOURS = 192
```

Input Rainfall CSV Format

date,precipitation

2013-01-15 01:00,0.076211111

- File used: seminar/precipitation.csv
- Single-basin rainfall series

Updating Simulation Window

- Update Start/End in `.gage`
- Update Start/End in `.control`
- Read model time step (`Time Interval`) from `.control`

Code references:

- `update_gage_times`
- `update_control_times`

Writing Rainfall to DSS

- Read CSV and align to simulation period
- Resample rainfall to model interval
- Write to DSS with `TimeSeriesContainer`
- It will be written into the DSS file and the corresponding path specified in the `.gage` file:
 - `parse_precip_pathname`
 - `load_precip_csv`
 - `write_precip_to_dss`

Running HEC-HMS in Batch Mode

- Create small Python script (`script_run`)
- Call `HEC-HMS.cmd -s script_run`
- Run without GUI interaction

Code reference: `run_hec_hms`

Reading Results from Output DSS

- Query flow pathnames for:
 - selected run (`RUN_NAME`)
 - selected element (`TARGET_HMS_ELEMENT = End_Point`)
- Read selected time window
- Export CSV with two columns:
 - `simulated_datetime`
 - `flow_m3s`

Code reference: `read_results`

Live Demo: Open file in Spyder

1. Open Anaconda Navigator
2. Launch Spyder from `hms_standalone` environment
3. Set working directory to `seminar/`



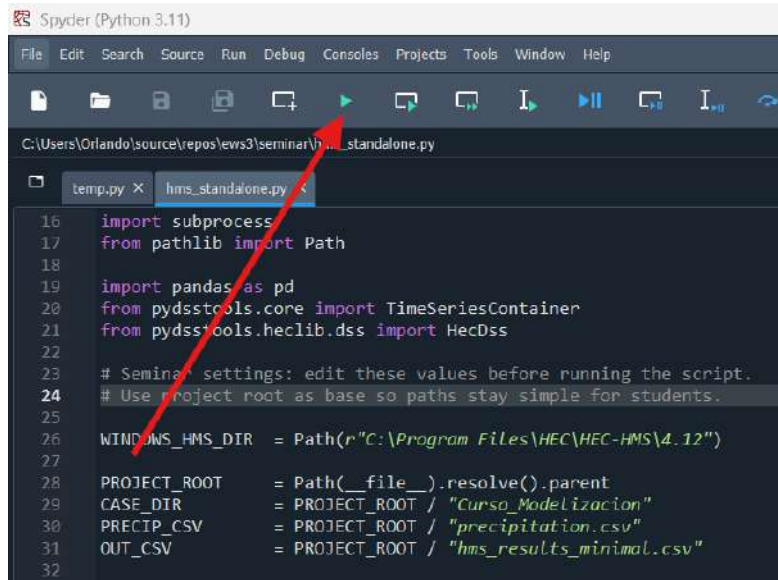
4. Open `hms_standalone.py`

Live Demo: Edit Config in Spyder

1. Open `seminar/hms_standalone.py`
2. Edit the parameters to match your HEC-HMS case
 - `RUN_NAME` "Run 1" (check for spaces/undercores...)
 - `TARGET_HMS_ELEMENT` `End_Point`
 - `START_UTC` `15 January 2013 1:00`
 - `SIMULATION_HOURS` `192 (8 days)`
3. Confirm paths:
 - `CASE_DIR`
 - `PRECIP_CSV`
 - `WINDOWS_HMS_DIR`

Live Demo: Run in Spyder

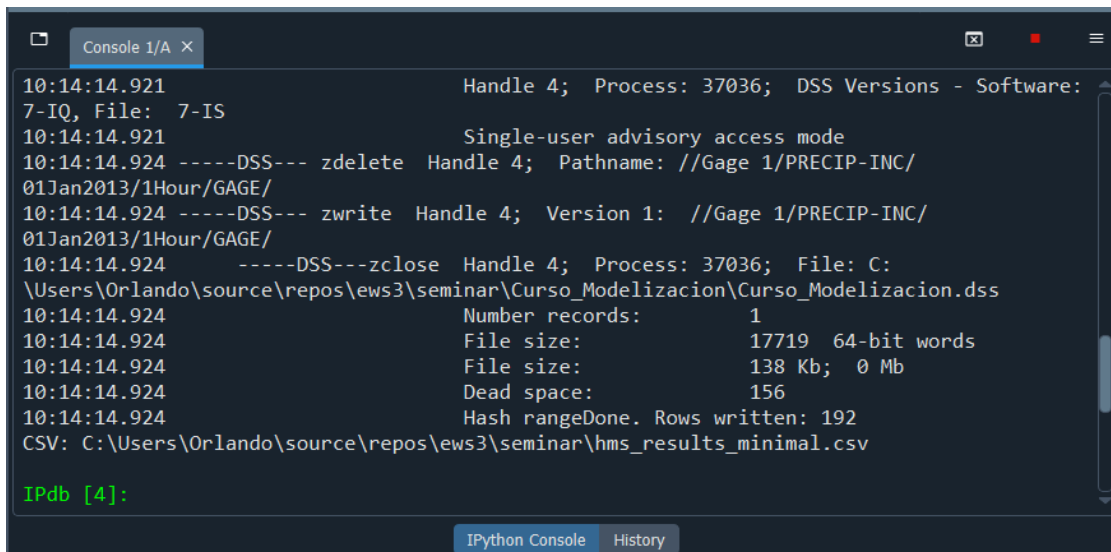
Press **F5** (Run File)



```
16 import subprocess
17 from pathlib import Path
18
19 import pandas as pd
20 from pydsstools.core import TimeSeriesContainer
21 from pydsstools.heclib.dss import HecDss
22
23 # Seminar settings: edit these values before running the script.
24 # Use project root as base so paths stay simple for students.
25
26 WINDOWS_HMS_DIR = Path(r"C:\Program Files\HEC\HEC-HMS\4.12")
27
28 PROJECT_ROOT = Path(__file__).resolve().parent
29 CASE_DIR = PROJECT_ROOT / "Curso_Modelizacion"
30 PRECIP_CSV = PROJECT_ROOT / "precipitation.csv"
31 OUT_CSV = PROJECT_ROOT / "hms_results_minimal.csv"
32
```

Live Demo: Check Output

Check the Console for any errors



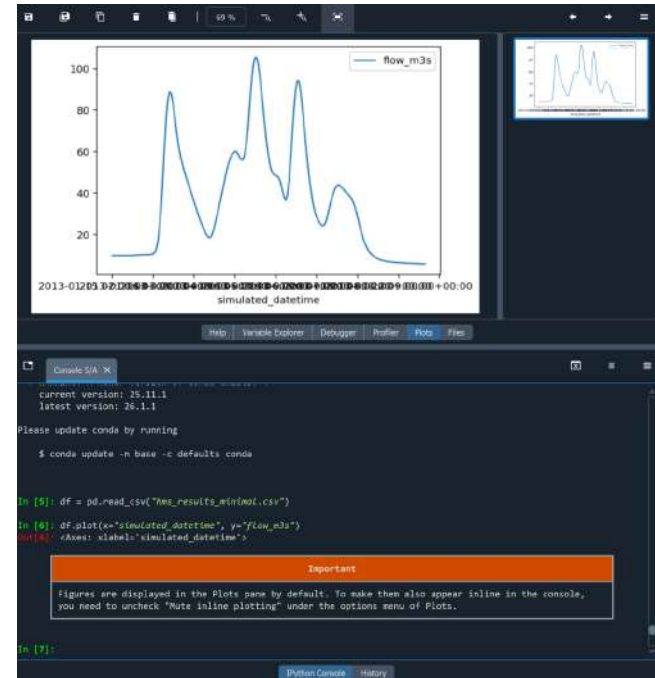
```
10:14:14.921 Handle 4; Process: 37036; DSS Versions - Software:
7-IQ, File: 7-IS
10:14:14.921 Single-user advisory access mode
10:14:14.924 -----DSS--- zdelete Handle 4; Pathname: //Gage 1/PRECIP-INC/
01Jan2013/1Hour/GAGE/
10:14:14.924 -----DSS--- zwrite Handle 4; Version 1: //Gage 1/PRECIP-INC/
01Jan2013/1Hour/GAGE/
10:14:14.924 -----DSS---zclose Handle 4; Process: 37036; File: C:
\Users\Orlando\source\repos\ews3\seminar\Curso_Modelizacion\Curso_Modelizacion.dss
10:14:14.924 Number records: 1
10:14:14.924 File size: 17719 64-bit words
10:14:14.924 File size: 138 Kb; 0 Mb
10:14:14.924 Dead space: 156
10:14:14.924 Hash rangeDone. Rows written: 192
CSV: C:\Users\Orlando\source\repos\ews3\seminar\hms_results_minimal.csv

IPdb [4]:
```

Live Demo: Inspect Output in Spyder

1. Open `seminar/hms_results_minimal.csv`
2. Verify columns:
 - `simulated_datetime`
 - `flow_m3s`
3. Plot quick hydrograph in console by running:
 - `conda install matplotlib`
 - `df = pd.read_csv("hms_results_minimal.csv")`
 - `df.plot(x="simulated_datetime", y="flow_m3s")`

Note: installing matplotlib takes a few minutes.



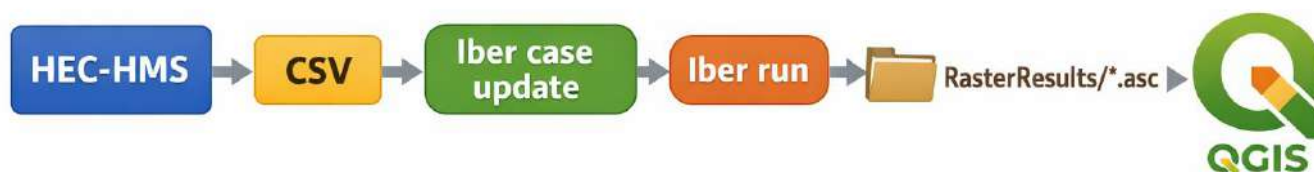
Connection to Full EWS

- This script is intentionally minimal.
- Production EWS adds:
 - automatic data retrieval
 - quality checks
 - scheduling
 - alert generation

Second Session Scope

- Build on the previous HEC-HMS session output.
- Use one prepared hydraulic case: `Sarria.gid`.
- Use one prepared hydrograph file: `hms_results_minimal.csv`.
- Focus on:
 - Which Iber files matter.
 - How to convert a built case into a reusable template-style case.
 - How the minimal Python script injects flow and runs Iber.
- Out of scope:
 - Calibration
 - Optimization
 - Data retrieval
 - Plotting automation.

End-to-End Pipeline



- HEC-HMS provides discharge time series.
- Iber uses those series as inlet boundary conditions.
- Iber outputs rasters for depth/velocity that can be visualized in a GIS software.

Material for the session

- `Sarria.gid` (Iber case built in the previous sessions)
- `hms_results_minimal.csv` (hydrograph computed in previous session)
- `run_iber_case.py` (minimal automation script)

Input CSV

- We will use the output CSV from the previous session.
- Required CSV columns for this session:
 - `simulated_datetime`
 - `flow_m3s`
- File used:
`hms_results_minimal.csv`

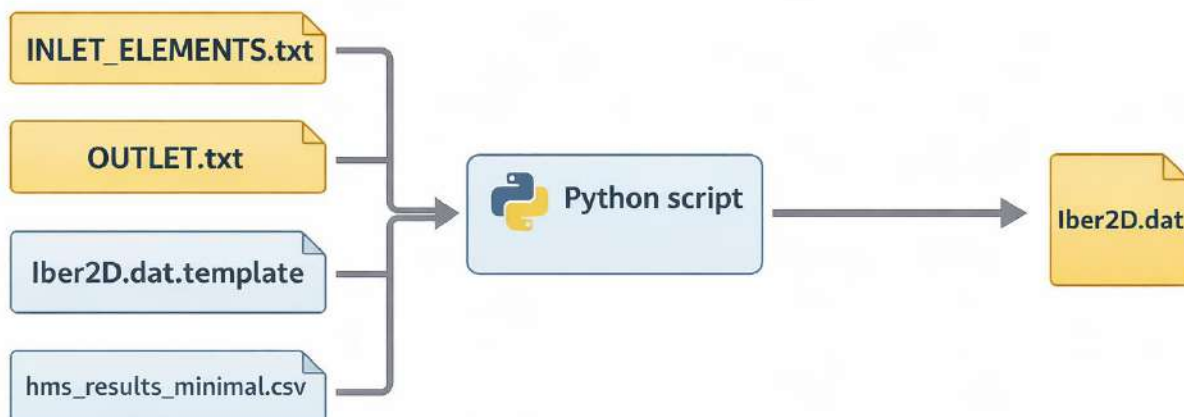
```
simulated_datetime,flow_m3s
2013-01-15 02:00:00+00:00,10.0
2013-01-15 03:00:00+00:00,9.999803
2013-01-15 04:00:00+00:00,9.998819
2013-01-15 05:00:00+00:00,9.99659
2013-01-15 06:00:00+00:00,9.993394
2013-01-15 07:00:00+00:00,9.989946
2013-01-15 08:00:00+00:00,9.986757
2013-01-15 09:00:00+00:00,9.984022
2013-01-15 10:00:00+00:00,9.982395
2013-01-15 11:00:00+00:00,9.983965
2013-01-15 12:00:00+00:00,9.992073
2013-01-15 13:00:00+00:00,10.010147
2013-01-15 14:00:00+00:00,10.0408325
2013-01-15 15:00:00+00:00,10.0854
2013-01-15 16:00:00+00:00,10.142466
2013-01-15 17:00:00+00:00,10.206963
2013-01-15 18:00:00+00:00,10.270849
2013-01-15 19:00:00+00:00,10.325048
2013-01-15 20:00:00+00:00,10.361566
2013-01-15 21:00:00+00:00,10.376524
2013-01-15 22:00:00+00:00,10.374927
2013-01-15 23:00:00+00:00,10.372664
2013-01-16 00:00:00+00:00,10.39238
2013-01-16 01:00:00+00:00,10.456919
2013-01-16 02:00:00+00:00,10.600714
2013-01-16 03:00:00+00:00,10.920757
2013-01-16 04:00:00+00:00,11.681981
2013-01-16 05:00:00+00:00,13.44275
2013-01-16 06:00:00+00:00,17.266493
2013-01-16 07:00:00+00:00,24.432463
2013-01-16 08:00:00+00:00,35.361797
```

Iber Case Anatomy

- A `.gid` folder contains many files.
- The `.dat` files are the **input files** for the numerical solver.
- We will **reutilize** the `.dat` files generated in the Iber GUI and **modify only the boundary conditions** located in the `Iber2D.dat` file.
- In this workflow, we will create the following files:
 - `Iber2D.dat.template` (`Iber2D.dat` file without the boundary conditions section)
 - `INLET_ELEMENTS.txt` (list of inlet boundaries: `element_id element_side`)
 - `OUTLET.txt` (outlet boundary lines kept unchanged)
 - `Iber2D.dat` (will be generated automatically by the script)
 - Output rasters: `RasterResults/* .asc`

Why Build a Template Case

- Separate static model data from dynamic event data.
- Avoid manual editing of long hydrograph blocks each run.
- Enable fast re-runs with new hydrograph CSV files.



The Section We Modify

- In Iber2D.dat, boundary conditions are located after the line:
 - CC: CONDICIONS CONTORN
- Inlet hydrograph lines contain:
 - -Hidrograma Q CriticoLento-
- Outlet lines must remain **unchanged** and contain:
 - -Salida en Critico/Rapido-

```
CC: CONDICIONS CONTORN
87 1 0 -29.437716 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 3600 -32.882885 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 7200 -36.451020 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 10800 -40.160927 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 14400 -43.912000 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 18000 -47.772877 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 21600 -51.436500 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 25200 -54.349380 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 28800 -56.618607 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 32400 -58.367104 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 36000 -59.495983 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 39600 -60.156754 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 43200 -59.790400 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 46800 -58.551907 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 50400 -57.052315 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 54000 -56.047190 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 57600 -56.347427 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 61200 -59.018040 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 64800 -64.408950 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 68400 -72.743930 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 72000 -83.292370 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 75600 -93.425420 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 79200 -101.125250 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 82800 -105.080900 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 86400 -105.528940 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 90000 -102.814840 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 93600 -97.488030 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 97200 -89.989400 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 100800 -81.731590 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 104400 -73.625270 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 108000 -66.330750 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 111600 -60.129770 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
...
69113 2 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103399 1 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103403 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103473 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103475 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103481 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103485 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103499 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103500 2 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103502 2 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
105505 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
105941 1 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
107205 1 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
123456789
```

The Section We Modify

- The inlet lines column format:
 1. Mesh element ID
 2. Mesh element side
 3. Time
 4. Flow in m³/s
 5. ...
- The inlet time series is repeated for each boundary edge!!
- To create the INLET_ELEMENTS.txt we need to obtain a list of:
 - element_id element_side
- We need to simplify the **inlet** with a time series **with a single timestep**.

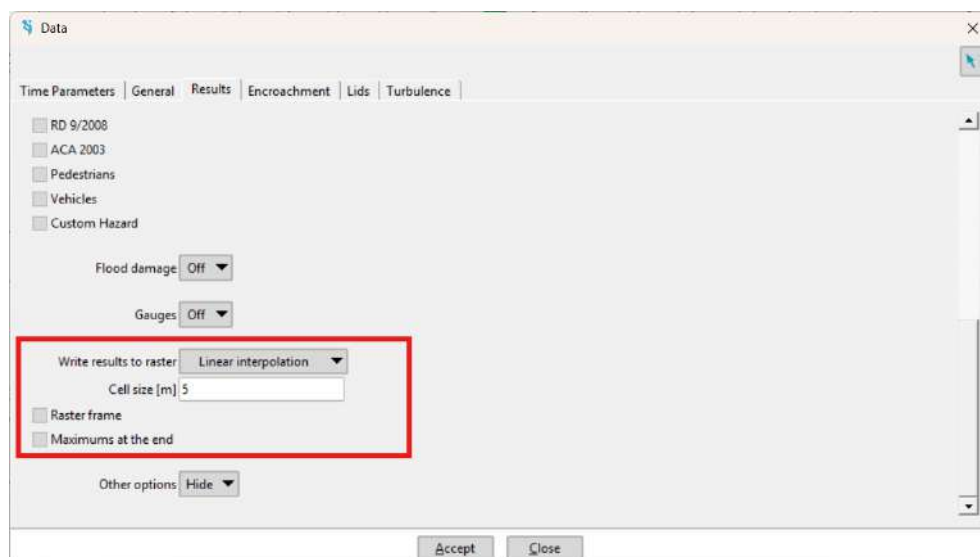
```
87 1 0 -29.437716 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 3600 -32.882885 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 7200 -36.451020 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 10800 -40.160927 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 14400 -43.912000 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 18000 -47.772877 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 21600 -51.436500 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 25200 -54.349380 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 28800 -56.618607 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 32400 -58.367104 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 36000 -59.495983 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 39600 -60.156754 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 43200 -59.790400 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 46800 -58.551907 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 50400 -57.052315 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 54000 -56.047190 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 57600 -56.347427 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 61200 -59.018040 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 64800 -64.408950 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 68400 -72.743930 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 72000 -83.292370 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 75600 -93.425420 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 79200 -101.125250 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 82800 -105.080900 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 86400 -105.528940 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 90000 -102.814840 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 93600 -97.488030 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 97200 -89.989400 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 100800 -81.731590 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 104400 -73.625270 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 108000 -66.330750 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
87 1 111600 -60.129770 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
```

Adapting the Iber case

- Open the Sarria case in Iber
- Save the case into the seminar folder: Files → Save as...
- Now we will enable the raster output and simplify the inlet...

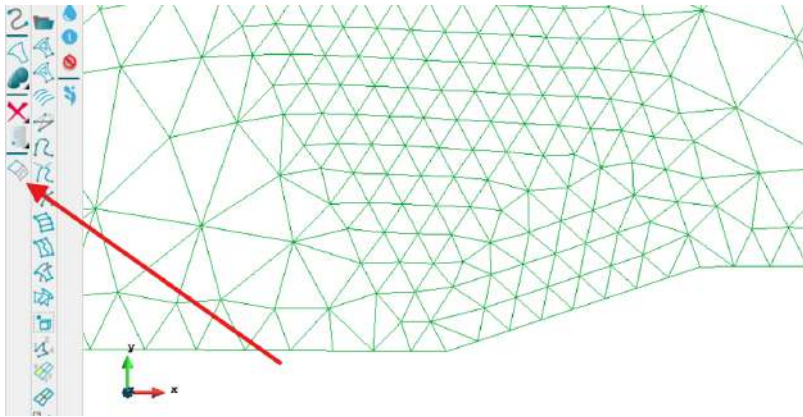
Adapting the Iber case

- To enable the raster output:
 - Data → Problem Data... → Results → Write results to raster → Linear interpolation



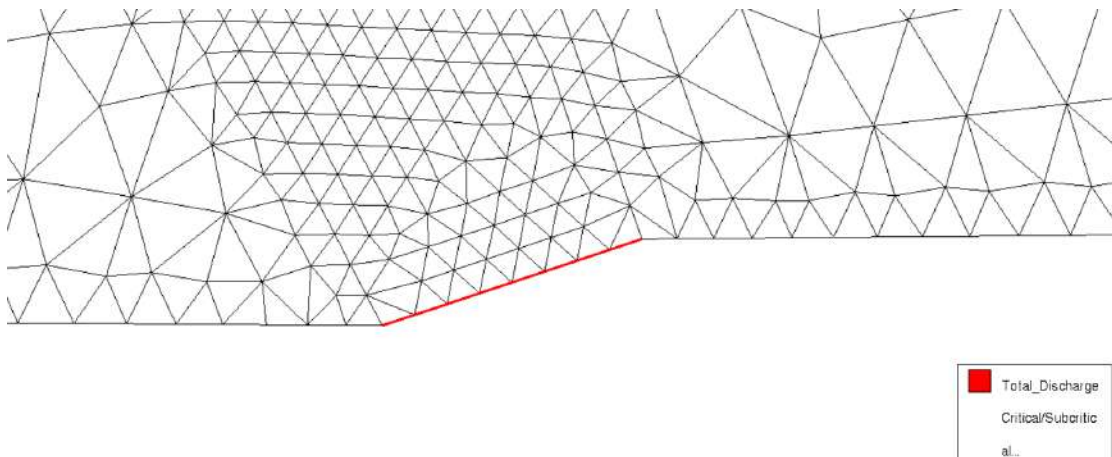
Adapting the Iber case

- We need to simplify the **inlet** with a time series **with a single timestep**.
- For this course we will assign the **new inlet boundary conditions** directly **into the mesh** to avoid remeshing, assign elevation...
- In preprocessing, switch to mesh view.



Adapting the Iber case

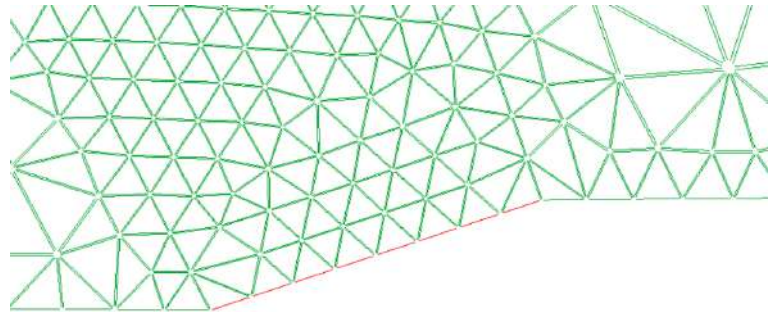
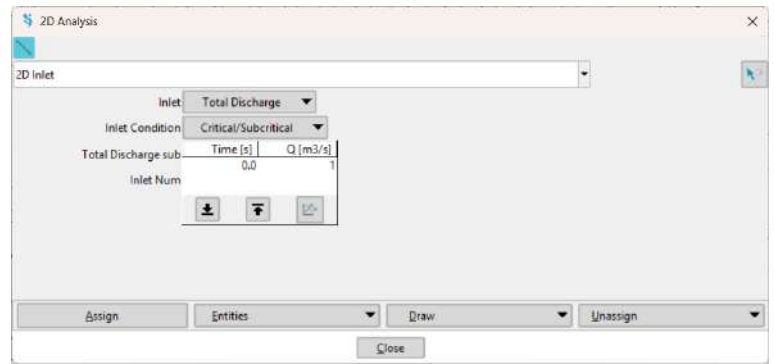
- Data → Hydrodynamics → Boundary Conditions... → Draw → Colors



- We will assign new inlet boundary conditions to these edges!

Adapting the Iber case

- Create new inlet boundary condition with a constant flow of $1 \text{ m}^3/\text{s}$
- Click on Assign and click on the inlet edges, then press Finish or Esc.



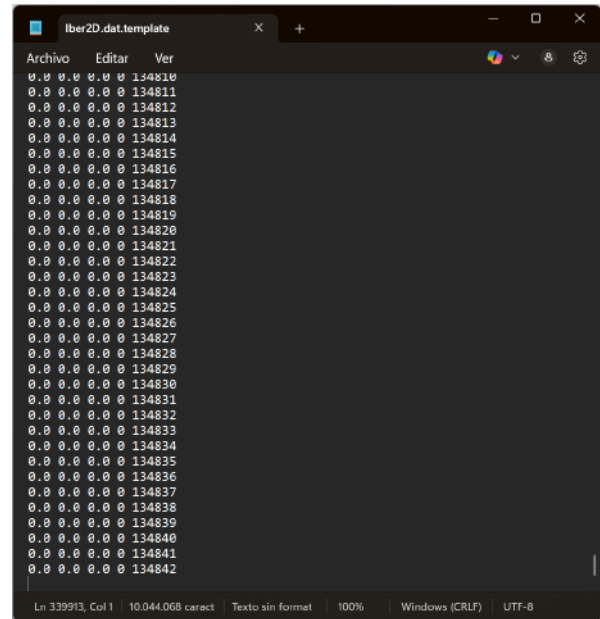
Adapting the Iber case

- Now run the case to re-generate the .dat files with Calculate → Calculate
- You can cancel the process once the first timestep is calculated with Calculate → Cancel process
- Check that the boundary conditions section of Iber2D.dat is now much more compact.

```
CC: CONDICIONS CONTORN
87 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
94 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
100 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
106 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
112 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
119 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
125 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
131 1 0 -1 -1 1 -41 2 1 1 -Hidrograma Q CriticoLento-
69113 2 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103399 1 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103403 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103473 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103475 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103481 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103485 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103499 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103500 2 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
103502 2 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
105505 3 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
105941 1 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
107265 1 0 0 0 0 -40 1 1 0 1 -Salida en Critico/Rapido -
123456789
```

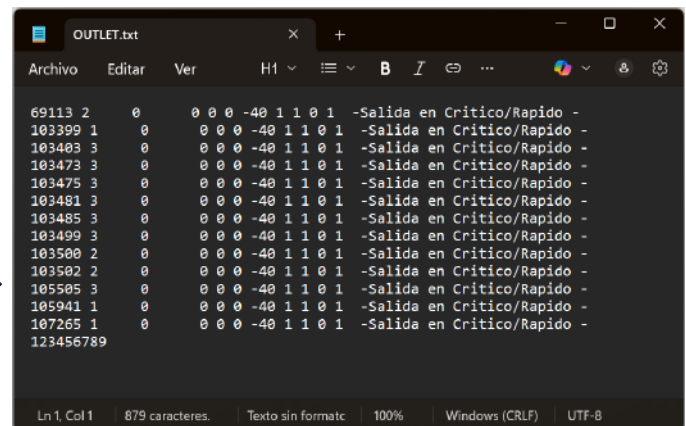
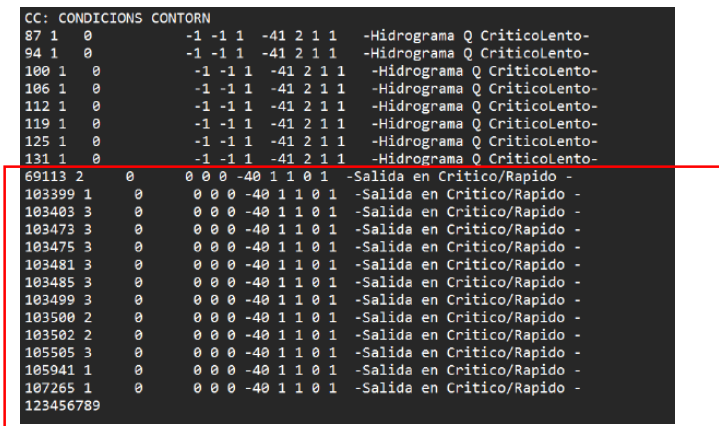
Create the template files

- Open the Iber2D.dat with the notepad and find the line:
 - CC: CONDICIONS CONTORN
- Delete that line and all the subsequent lines.
- Leave a blank line at the end.
- Save the file as Iber2D.dat.template



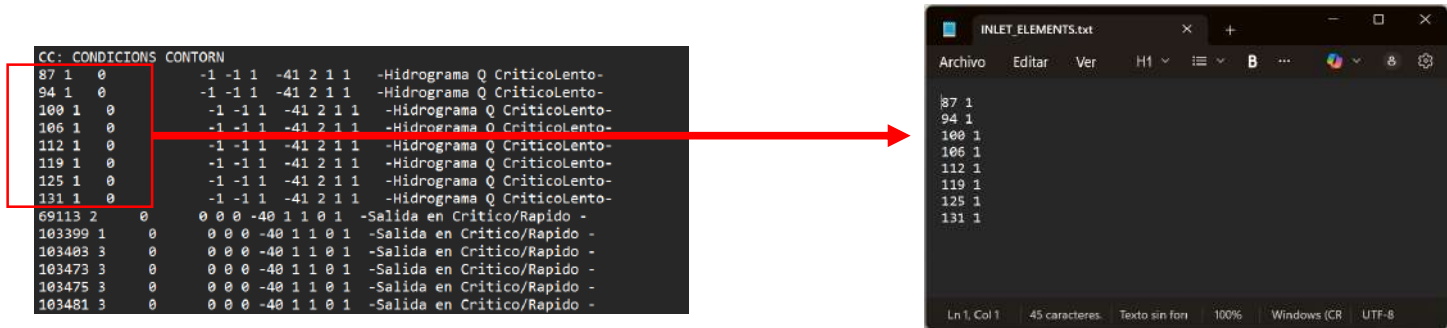
Create the template files

- Open the Iber2D.dat with the notepad and find the line:
 - CC: CONDICIONS CONTORN
- Copy the outlet boundary lines and paste them in a new file.
- Save the file as OUTLET.txt



Create the template files

- Open the `Iber2D.dat` with the notepad and find the line:
 - `CC: CONDICIONS CONTORN`
- Copy the inlet boundary lines and paste them in a new file.
- Keep only the first two columns (element ID and element side).
- Save the file as `INLET_ELEMENTS.txt`



Script `run_iber_case.py`: Structure Overview

- `load_flow_values`: read and validate flow CSV.
- `read_inlet_boundaries`: load inlet boundaries from `INLET_ELEMENTS.txt`.
- `generate_hydrograph_lines`: create discharge lines for each boundary and time step.
- `build_iber2d`: copy template and append hydrograph + outlets.
- `run_iber`: execute Iber with configured parameters.
- Final check: count `.asc` rasters in `RasterResults`.

Config Block Walkthrough

- Check the paths and set the simulation parameters.

```
27 # -----
28 # EDIT THESE VALUES BEFORE RUNNING THE SCRIPT.
29 # Students only need to modify this block.
30 # -----
31 CASE_DIR = Path("Sarria.gid")
32 FLOW_CSV = Path("hms_results_minimal.csv")
33 IBER_EXE = Path(r"C:/Program Files/Iber/Iber 3.4/problemtypes/IBER.gid/bin/windows/IberPlus.exe")
34
35 # Simulation settings
36 WARM_SECONDS = 6 * 3600
37 MAX_TIME_SECONDS = 30 * 3600
38 WRITE_SECONDS = 600
39
40 # csv settings
41 CSV_DT_SECONDS = 3600
42 START_DATETIME = "2013-01-17 18:00"
43 # -----
```

Config Block Walkthrough

- **WARM_SECONDS** : time after which results begin to be written (useful to initialize the river channel).
- **MAX_TIME_SECONDS** : maximum time of the simulation (including WARM_SECONDS).
- **WRITE_SECONDS** : raster writing time step.
- **CSV_DT_SECONDS** : time step used in the input CSV file.
- **START_DATETIME** : datetime from which the flow time series is extracted.

CSV Load and Validation

- `load_flow_values`: function that reads and validates flow CSV.
- `pandas.read_csv` uses fixed separator `,` and `utf-8-sig` encoding.
- Script validates:
 - required columns exist
 - `simulated_datetime` is parseable as datetime
 - `flow_m3s` values are numeric
 - rows exist after filtering from `START_DATETIME`

Case Update Logic

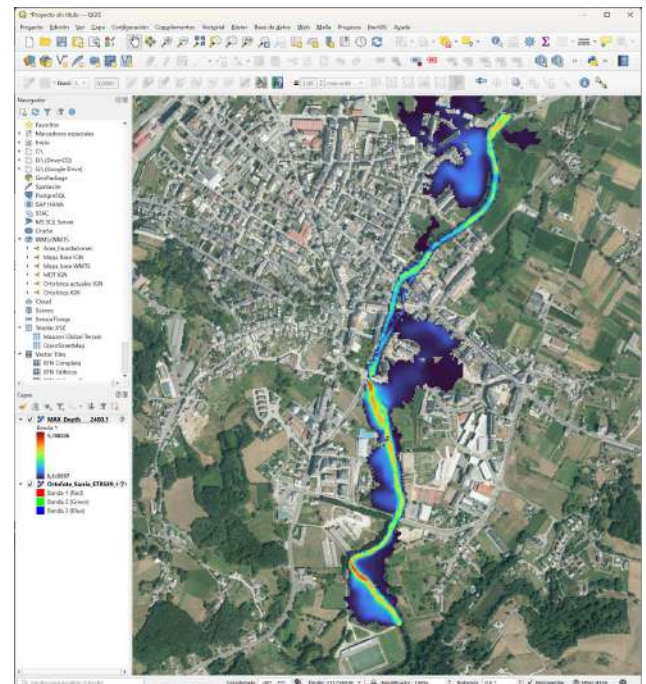
- Copy `Iber2D.dat.template` to `Iber2D.dat`.
- Append `CC: CONDICIONS CONTORN`.
- Append generated hydrograph lines for all inlet boundaries.
- Append contents of `OUTLET.txt`.

Iber Execution Command

- Key flags:
 - `-r`: warm-up (seconds)
 - `-m`: maximum simulation time (seconds)
 - `-d`: case directory
 - `-w`: raster write interval (seconds)
 - `--noasciireresults --nogidresults`

Output and Quick Validation

- Rasters are kept in the case folder.
- Check after run:
 - `Sarria.gid/RasterResults/*.asc` exists.
- These can be opened in QGIS for visual review.



Live Demo Sequence

1. Open `run_iber_case.py` in Spyder
2. Confirm config paths and variables.
3. Confirm the CSV has columns `simulated_datetime` and `flow_m3s`.
4. Run script.
5. Verify `.asc` raster files in `Sarria.gid/RasterResults`.
6. Open one raster in QGIS.

Troubleshooting and Wrap-Up

- Common issues:
 - wrong `IBER_EXE` path
 - missing CSV column (`flow_m3s`)
 - non-numeric or empty CSV values
 - missing `Iber2D.dat.template`, `INLET_ELEMENTS.txt`, or `OUTLET.txt`
- Key takeaway:
 - Hydrology output can feed hydraulic modeling automatically with minimal code.

Interreg



Cofinanciado por
la Unión Europea
Cofinanciado pela
União Europeia

RISC_PLUS

España - Portugal

risc-plus.eu



Universidade de Vigo



El proyecto 0031_RISC_PLUS_6_E está cofinanciado por el Fondo Europeo de Desarrollo Regional FEDER en el marco del programa Interreg VI A España - Portugal (POCTEP) 2021-2027