

# 1 Descripción del flujo del código

El modelo desarrollado simula un sistema de seis balsas de evaporación conectadas secuencialmente, utilizando PHREEQC como motor de cálculo geoquímico. La ejecución completa se coordina desde los scripts `run_simulation.bat` (local, Windows) o `run_portable.bat` (versión portable), los cuales lanzan la simulación principal mediante la instrucción:

```
python -m src.run --plot
```

## 1.1 Configuración inicial

El archivo de configuración `config.yaml` contiene todas las rutas y parámetros necesarios para ejecutar la simulación:

- Rutas al ejecutable de PHREEQC y a su base de datos geoquímica.
- Rutas a los ficheros de entrada (`brineData.txt`, `pondsData.txt`, `evap_diaria.csv`).
- Directorio de trabajo donde se almacenan los resultados (`experiment_results/`).
- Parámetros físicos: volumen inicial de la balsa 1 y densidad del líquido.

Durante la ejecución, el módulo `src/utils/config.py` carga esta configuración y la pone a disposición del resto de componentes.

## 1.2 Carga de datos de entrada

La lectura de los archivos de entrada se realiza en el módulo `src/io/loaders.py`:

- **`brineData.txt`**: composición inicial de la salmuera en formato `SOLUTION` de PHREEQC.
- **`pondsData.txt`**: listado de balsas y sus volúmenes máximos (en  $\text{m}^3$ ).
- **`evap_diaria.csv`**: tasas diarias de evaporación (mol  $\text{H}_2\text{O}$  por litro y día).

Si no se proporciona un archivo de evaporación, el sistema emplea una tasa constante definida en el propio archivo de configuración.

## 1.3 Inicialización del modelo

La estructura básica del sistema se define en `src/domain/models.py`, que contiene las clases:

- **`Pond`**: representa una balsa individual.
- **`Plant`**: agrupa todas las balsas y define el sistema completo.
- **`SimulationParams`**: almacena los parámetros de simulación.
- **`MineralProps`**: gestiona las propiedades de las fases minerales relevantes.

La comunicación con PHREEQC se gestiona desde el módulo `src/domain/phreeqc_runner.py`, que utiliza el envoltorio `phreeqcModel()` definido en `workingTools.py`. Este módulo genera los archivos `input.in` y `output.out`, y ejecuta el binario de PHREEQC con las condiciones especificadas.

## 1.4 Ejecución principal

El flujo principal se implementa en `src/run.py`. Este módulo:

1. Crea una instancia de `PhreeqcRunner`.
2. Carga los datos de entrada y configura la simulación.
3. Inicia el proceso mediante la función `run_full_pipeline()` de la clase `Simulation`.

Durante este proceso, se generan sucesivamente los archivos de resultados `results.dat`, `results2.dat`, `results3.dat`, etc., hasta completar las seis balsas.

## 1.5 Lógica de simulación y transferencia

El método `run_full_pipeline()` replica el comportamiento físico del sistema de balsas:

1. La **balsa 1** inicia la concentración de la salmuera mediante evaporación diaria.
2. Cuando comienza la **precipitación de halita**, detectada por la función `find_transfer_day_halite()`, se determina el día de transferencia.
3. A partir de los datos de evaporación acumulada se calcula el volumen restante de líquido:

$$V_{\text{restante}} = n_{\text{agua}} \times \frac{18.01528}{\rho_{\text{líquido}} \times 10^{-3} \text{ donde}}$$

$\rho_{\text{líquido}}$  es la densidad (g/L) definida en `config.yaml`.

4. Se compara el volumen restante con la capacidad de la siguiente balsa (definida en `pondsData.txt`).
  - Si el volumen cabe, se transfiere completamente.
  - Si no cabe, solo se transfiere la fracción posible y el exceso se descarta.
5. El evento se imprime en consola, por ejemplo:

```
[TRANSFER CAPACITY] pond1 -> pond2 | requested=25000 m³ | allowed=15000 m³ | discarded=10000 m³
```

6. El proceso se repite secuencialmente para las seis balsas (Pond 1 → Pond 6).

## 1.6 Cálculo de evaporación

Cada fila del archivo `evap_diaria.csv` representa un día de simulación y un paso de reacción en PHREEQC. La reacción de evaporación se define mediante bloques **REACTION** en el archivo de entrada, del tipo:

```
REACTION 1
Water
-0.1938 mol
-0.3973 mol
...
INCREMENTAL_REACTIONS true
```

Si no existe un archivo de evaporación, se aplica una tasa constante definida en los parámetros de simulación (`evaporation_rate_mol_per_day_L`).

## 1.7 Resultados y análisis

Los resultados se almacenan en el directorio `experiment_results/` con la siguiente estructura:

- `output/`: archivos de salida de PHREEQC (`results*.dat`).
- `plots/`: gráficos generados automáticamente si se usa la opción `--plot`.
- `run_summary/`: informes resumen de transferencias y evolución del sistema.

Los gráficos se generan mediante las funciones del módulo `src/utils/plots.py`, que muestran la evolución de las fases minerales (halita, yeso y calcita) en cada balsa, así como comparativas entre la evolución de la balsa 1 y las sucesivas.

Durante la ejecución, se imprime por consola un resumen cronológico de los eventos principales: inicio y fin de cada simulación, día de transferencia, volumen restante, capacidad utilizada y excedente descartado.

## 1.8 Resumen del flujo completo

1. Se ejecuta `run_simulation.bat`.
2. Se carga la configuración desde `config.yaml`.
3. Se leen los archivos de entrada de salmuera, balsas y evaporación.
4. Se inicializa el modelo y se ejecuta la simulación de la balsa 1.
5. Se detecta la formación de halita y se realiza la transferencia parcial o completa.
6. El proceso se repite secuencialmente hasta la balsa 6.
7. Se generan los resultados, gráficos e informes finales.

Este flujo permite simular de forma realista el comportamiento de un sistema de balsas de evaporación, considerando tanto las condiciones geoquímicas de la salmuera como las restricciones físicas de capacidad y las variaciones estacionales de evaporación.