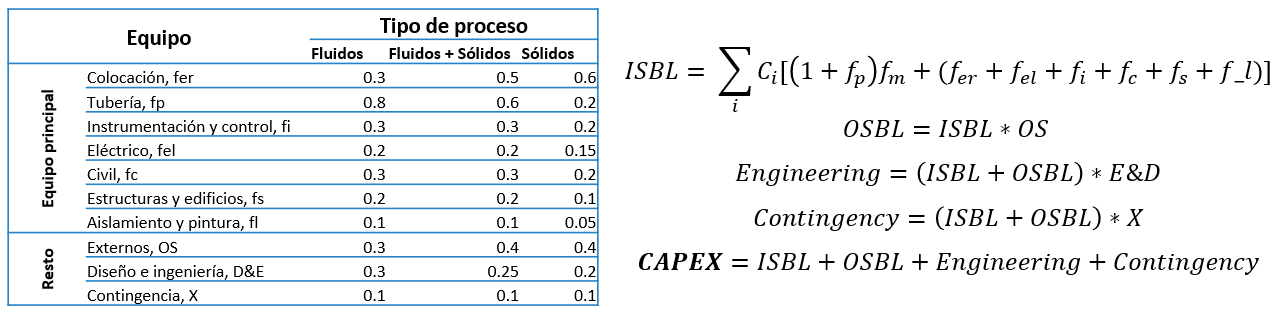
# **Prueba desarrollador Python junior**

## Introducción

Una empresa de consultoría realiza análisis tecno-económicos de procesos industriales para sus clientes. Para ello utiliza un método detallado de estimación factorial basado en [factores de Lang](https://chemicalprojects.wordpress.com/tag/lang-factor/). El cual permite, dado el coste de un equipo en concreto, estimar el coste total de su instalación (canalizaciones, electricidad, instrumentación…) y los costes indirectos (ingeniería, diseño…) de la siguiente forma:



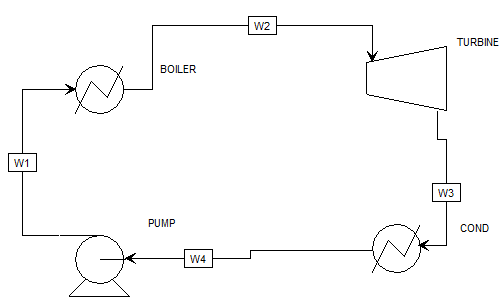
Para el precio de los equipos, se basa en datos privados o bibliográficos y/o correlaciones de equipos. Cuando se dispone de un dato de coste a una determinada capacidad, éste se estima siguiendo la correlación de William:

Dónde:

* Cost == Coste a estimar
* Cost\_ref == Coste de referencia (dato)
* Cap == Capacidad del coste a estimar
* Cap\_ref == Capacidad de referencia (dato)
* n == factor entre 0,6 – 1

Hasta la fecha, la consultora realiza el análisis apoyándose en scripts de Python funcionales para cada caso en concreto. Dado la repetición de casos de estudio y de líneas de código entre casos decide aplicar el paradigma de la programación orientada a objetos para crear una clase o clases que les facilite esta labor.

## Caso de estudio

Una empresa de revalorización de residuos contrata a la consultora para analizar la viabilidad económica de utilizar una corriente que actualmente deshecha en su proceso para producir electricidad.

Dicha corriente tiene la capacidad calorífica suficiente para producir **10,000 kg/h de vapor de agua a 70 bares** que pueden ser aprovechados en una **turbina de 1500 kW** para producir electricidad. Adicionalmente, serán necesarios tanto **una bomba que mueva un caudal de 2.84 L/s** y una **torre de refrigeración que condensará** el vapor que sale de la turbina.

Para la estimación del coste del generador de vapor (**boiler**), la turbina (**turbine**) y la bomba (**pump**) se utilizan **correlaciones bibliográficas**. Para la torre de refrigeración (**condenser**) se conoce de otro proyecto que el coste ya instalada y funcionando es de 400,000 € con una capacidad de 15,000 kg/h de tratamiento, por lo que se utilizará una **correlación de William** con n = 0.8.

El **factor de capacidad** de la instalación será del **90 %** (en un año funcionará el 90 % del tiempo ininterrumpidamente). El coste de **agua destilada** para la turbina se estima en **1.29 €/ton**, la **electricidad** se venderá a **50 €/MWh** y se estima la necesidad de **4 puestos de trabajo en 3 turnos con un salario medio de 30,000 €** (coste de empresa).

Para financiar el proyecto, la empresa de revalorización de residuos prevé pedir un **préstamo del 60 % de la inversión a 10 años con un tipo de interés del 4%**. Además, la **depreciación** aplicada a este tipo de plantas es del **7% anual**.

La empresa, pide a la consultora realizar una estimación de los **flujos de caja del proyecto a 20 años** junto con su valor actual neto (**VAN** o **NPV** en inglés) y su tasa interna de retorno (**TIR** o **IRR**), para valorar la rentabilidad del proyecto.

Junto a éste documento encontrarás el archivo “economic\_assessment.py” que corresponde a un script en Python que resuelve la estimación económica del caso de estudio anteriormente presentado. El script utiliza la programación funcional: Primero define los factores de Lang, luego las correlaciones para estimar el coste de los equipos como funciones que pueden estimar el coste del equipo en cuestión y si se quiere el coste del equipo listo para funcionar (installed = True) se utilizan los factores de Lang. El script define funciones para calcular el principal (el dinero que devuelves) y los intereses a pagar de un préstamo (lo que te cobran por el préstamo) y para calcular la [depreciación](https://es.wikipedia.org/wiki/Depreciaci%C3%B3n). Finalmente, utiliza estas funciones para calcular el [CAPEX](https://es.wikipedia.org/wiki/Capex), el [OPEX](https://es.wikipedia.org/wiki/Opex) y el modelo financiero a 20 años, incluyendo los beneficios antes de impuestos (EBT por sus siglas en inglés), los impuestos (30% sobre el EBT), el beneficio neto o después de impuestos (EAT), el flujo de caja ([cash flow](https://es.wikipedia.org/wiki/Flujo_de_caja)) y el flujo de caja acumulado; por último, utilizando una [tasa de descuento](https://es.wikipedia.org/wiki/Tasa_de_descuento) del 5.3%, calcula el [NPV](https://es.wikipedia.org/wiki/Valor_actual_neto) y el [IRR](https://es.wikipedia.org/wiki/Tasa_interna_de_retorno) del proyecto.

## Se pide

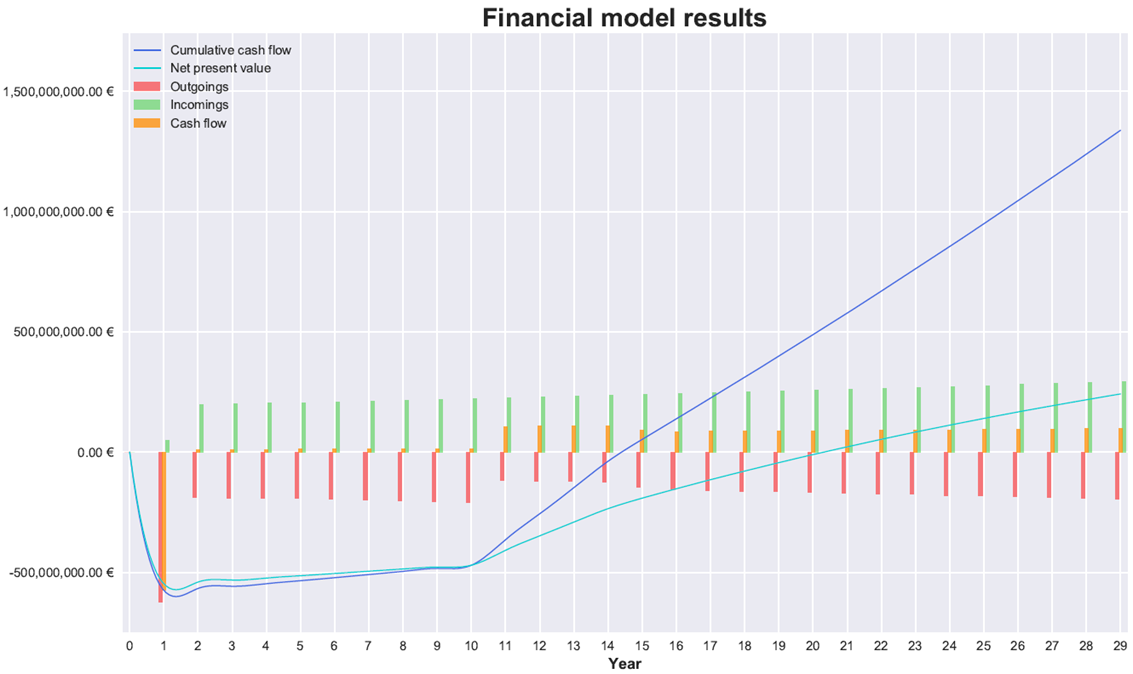
Esta resolución es análoga para todos los procesos industriales que la consultora analiza cambiando únicamente el coste de los equipos, por lo que se deben realizar los siguientes cambios:

* Utiliza las mejores prácticas de la programación orientada a objetos para crear una clase o clases que resuelvan el análisis económico de cualquier proceso dado.
* Debe incluir, como métodos de una clase general o de una “sub-clase” de equipos, las correlaciones de equipo dadas en “economic\_assessment.py” y la correlación de William (a cuyo resultado se le pueden aplicar también los factores de Lang) y ha de ser sencillo introducir nuevas correlaciones en el futuro.
* Debe incluir métodos para el cálculo del préstamo (loan()) y de la depreciación (depreciation()).
* Debe poder accederse como resultados a un dataframe como el del script (df), al NPV y al IRR.
* Las funciones del script actual pueden usarse tal cual o ser modificadas como se considere en los nuevos métodos desarrollados.
* Documenta la clase o clases desarrolladas en un archivo tipo README.md o un documento Word, preferiblemente en inglés.
* Resuelve el caso de estudio anterior utilizando la nueva clase o clases que has desarrollado.

## Adicionalmente

Las siguientes tareas son voluntarias:

* Incluir un método que calcule el [payback](https://es.wikipedia.org/wiki/Payback_(econom%C3%ADa)) de la inversión en función del flujo de caja acumulado, éste debe poder accederse como un resultado al igual que el NPV y el IRR.
* Incluir un método para visualizar las ventas, los costes, el flujo de caja y el flujo de caja acumulado en un gráfico similar al siguiente (se recomienda usar [matplotlib](https://matplotlib.org/)):



* Incluir un método que exporte los resultados a un archivo Excel.