

Guía del Flujo de Trabajo y Uso de Códigos Python

Esta guía explica el propósito de cada script de Python y el orden en que deben ser ejecutados para replicar los resultados del proyecto.

1. Generación de Escenarios de Variables de Incertidumbre

Esta es la primera etapa, donde se simulan los datos de entrada para el modelo. La idea es generar múltiples escenarios para las variables que no se conocen con certeza. Estos escenarios son cruciales para el análisis de riesgo.

- **VariableAI_Generacion_Central.py:** Este script genera escenarios de la **generación hidroeléctrica** de la central. Utiliza datos históricos para ajustar un Modelo de Mezcla Gaussiana (GMM) para cada estación del año y luego simula días típicos para crear escenarios a lo largo de X años. Se debe utilizar los informes técnicos de CAMMESA de postdespacho ya extraídos en formato Access.
- **VariableAI_CMO.py:** Se encarga de generar escenarios del **Costo Marginal Horario (CMO)**. Carga datos históricos de CMO, los preprocesa, los agrupa por estaciones y luego usa un método de "bootstrap directo" para generar 1000 escenarios de X años. Se debe utilizar el archivo Excel adjunto donde se encuentran todos los datos históricos de los CMO.
- **VariableAI_PagoporPotencia.py:** Este script simula el **Pago por Potencia**. Utiliza una distribución triangular (basada en un valor más probable y un porcentaje de incertidumbre) para generar 1000 escenarios para un horizonte de X años.
- **VariableAI_ConfiabilidadBESS.py:** Este código genera escenarios de la **disponibilidad del BESS (Sistema de Almacenamiento de Energía en Baterías)**. Simula fallas de los componentes (inversores, HVAC) y utiliza una distribución Beta para modelar la degradación y disponibilidad a lo largo de X años.

Flujo de trabajo: Debes ejecutar estos cuatro scripts en cualquier orden para generar los archivos CSV de entrada (escenarios_generacion_hidro.csv, escenarios_cmo.csv, etc.) que serán utilizados en las siguientes etapas.

2. Optimización del Modelo

Una vez que se tienen los escenarios de entrada, se procede a la etapa de optimización, donde se busca la configuración óptima del BESS para maximizar el valor del proyecto.

- **Optimizacion_MEM.py:** Este script realiza el proceso de optimización para el **Mercado Eléctrico Mayorista (MEM)** actual. Busca la potencia y duración óptimas del BESS para la operación conjunta con la central hidroeléctrica, considerando diferentes configuraciones del BESS y maximizando el VAN (Valor Actual Neto) y la TIR (Tasa

Interna de Retorno). Genera un reporte detallado en Excel con los resultados.

- **Optimizacion_MercadoLiberal.py:** Similar al script anterior, pero adaptado a un **mercado de energía liberalizado ficticio**. Este script maximiza los ingresos por arbitraje de energía y venta de servicios auxiliares, y también genera un reporte financiero detallado en Excel.

Flujo de trabajo: Primero debes ejecutar los scripts de la Etapa 1. Luego, puedes ejecutar Optimizacion_MEM.py para el análisis del mercado actual y Optimizacion_MercadoLiberal.py para el análisis del mercado liberalizado.

3. Análisis de Riesgo y Sensibilidad

Con los resultados de la optimización, se pueden realizar análisis más profundos para entender la robustez del proyecto frente a la incertidumbre.

- **Simulacion_Montecarlo.py:** Este script realiza un **Análisis de Riesgo con Simulación de Montecarlo**. Utiliza los escenarios de las variables generadas en la Etapa 1 y el modelo de optimización de mercado liberalizado para calcular el VAN y la TIR para cada escenario. Los resultados se presentan en histogramas y un reporte de Excel, mostrando la distribución de los resultados financieros y la probabilidad de éxito.
- **Analisis_Sensibilidad_MEM.py:** Este script realiza un **Análisis de Sensibilidad** en el marco MEM sobre las variables clave (por ejemplo, el CAPEX del BESS, el costo de O&M de la central, la tasa de descuento). Mantiene la configuración óptima del BESS fija y evalúa cómo los cambios en estas variables afectan el VAN y la TIR.

Flujo de trabajo: Después de ejecutar los scripts de optimización, puedes usar Simulacion_Montecarlo.py para una evaluación completa del riesgo. Por otro lado, Analisis_Sensibilidad_MEM.py te permitirá entender la influencia de variables específicas en la rentabilidad del proyecto.