

MOME – Modelación de Mercados Eléctricos

1. Análisis:

El proyecto “MOME” surge a partir de un modelo teórico sobre un mercado genérico de energía eléctrica, y está pensado en el desarrollo de una herramienta de uso experimental. El fin de este software es obtener información sobre los valores de algunos parámetros presentes en el modelo teórico, de los cuales se desconoce su valor real. El desarrollo de MOME se dio gracias a los estudios encontrados en la literatura sobre: mercados de commodities, expectativas adaptativas, expectativas racionales, mercados de futuros, simulación computacional, mercados de energía eléctrica desregulados, mecanismos de capacidad, y economía experimental. Además, gracias a que el proyecto fue desarrollado en lenguaje Python, aplicando las facilidades de una librería de libre uso (oTree), con este proyecto se aporta una herramienta gratuita a la comunidad científica, para permitir que pueda ser usada, modificada y mejorada en el futuro en el continuo avance de la ciencia en este tema.

Algunas consideraciones a tener en cuenta:

- Cada año (ronda) el participante tendrá que tomar decisiones de Inversión [unidades/año] en nuevas plantas, con el objetivo de maximizar la Utilidad Acumulada [\$] durante todo el experimento.
- La Utilidad Anual [\$] puede ser positiva o negativa (dependiendo del margen de utilidad que es la diferencia entre precio y costo de la energía).
- La Utilidad Acumulada [\$] es la suma de las Utilidades Anuales durante todo el juego y puede ser negativas (pérdida) o positivas (ganancia)
- Cada planta tiene un tiempo de construcción de 4 años. Es decir, desde el momento en que el participante decide Invertir en una nueva planta, esta se convierte en una capacidad en construcción. Esta empezará a producir pasados 4 años (rondas), tiempo después del cual deja de ser Capacidad en Construcción y pasa a ser Capacidad Instalada. Es decir, las capacidades en construcción no generan utilidades
- Las capacidades instaladas tienen un tiempo de vida útil de 16 años. Es decir, cada planta produce energía durante 16 años y luego sale del sistema. La Capacidad Instalada Total [unidades] no puede ser negativa, por lo que lo mínimo que se puede invertir es 0 [unidades].
- La suma de la capacidad instalada y construida por jugador no puede superar las 20 unidades, por tanto, la inversión no puede ser tal que se viole ese tope.
- El precio de la energía está en función de la capacidad total instalada del mercado y esta función es una línea recta con pendiente negativa e intercepto en el eje Y (precio de la energía) en 6 [\$/unidad] y en el eje X (capacidad instalada del mercado) en 60 [unidades]
- El Costo es constante y siempre es igual a 1 [\$/unidad].
- El total de las cinco compañías productoras de electricidad (participantes) inician con las mismas condiciones. Durante todos los años anteriores (al inicio de la partida), cada firma ha invertido 0.6875 [unidades/año], que es igual al resultado de dividir 11 unidades en el tiempo de vida útil de cada planta (16 años).

2. Diseño:

Para realizar el proyecto se tuvieron dos etapas. En la primera etapa se formuló un modelo matemático para simular el comportamiento del mercado de energía. En la segunda parte del proyecto se decidió llevar el modelo al mundo real por medio de un experimento que consta, de grupos de 5 participantes los cuales durante 70 rondas deben tomar decisiones para generar utilidades traducibles en dinero en el mundo real.

El software tiene en cuenta la estrategia individual y grupal que afectan el precio de la energía, por lo que tiene sus propias reglas de negocios que hacen que sea un experimento no genérico. A continuación se detalla la composición del experimento en cuanto a interfaces de usuario y funcionamiento en general.

Vistas del software

- Bienvenida: En esta vista se le explica al usuario el tipo de actividad, la recompensa, y las reglas del juego.

Bienvenido al experimento.

Bienvenido, usted va a participar en un juego de toma de decisiones que llamaremos experimento. Al final de esta sesión usted recibirá un pago en efectivo por su participación; todos los jugadores podrán recibir pagos diferentes, ya que el pago final dependerá de las decisiones que tome y del desempeño del jugador durante el juego.

Usted participará en este juego a través del computador en que está sentado, es importante que usted no hable y no se comunique con otros participantes durante el experimento. Igualmente, está prohibido el uso de celulares u otro dispositivo electrónico durante todo el tiempo que dure esta sesión. Si usted incumple alguna de estas reglas se le pedirá que abandone el juego y no recibirá pago alguno.

- Instrucciones: Aquí se explica el rol, el objetivo, y las normas que rigen el juego.

Instrucciones

Instrucciones del juego

Rol:
En este juego usted tomará el rol de una **compañía productora de electricidad**. Usted hace parte de un mercado eléctrico compuesto por **cinco participantes**, pero usted no sabe quiénes son los **otros cuatro jugadores** que hacen parte de su **mercado**, ni cómo operan ellos de forma individual.

Objetivo:
Cada año usted tendrá que tomar decisiones de **Inversión [unidades/año]** en nuevas plantas, con el **objetivo de maximizar la Utilidad Acumulada [S]** durante todo el experimento.

Utilidades:
La **Utilidad Anual [S]** puede ser positiva o negativa y se calcula de la siguiente manera:
$$\text{Utilidad Anual} = \text{Margen de utilidad} * \text{Capacidad Instalada Total}$$

Donde toda la **Utilidad Acumulada [S]** es la suma de las **Utilidades Anuales** durante toda el tiempo en la experiencia.

- Inversión: En esta pantalla el usuario debe ingresar por teclado la inversión que desea hacer para la capacidad en construcción del año actual. En esta pantalla se presenta información sobre la capacidad en construcción e instalada el participante; e información general del mercado y de los demás

- Primero empieza con una información inicial de cada participante, en base a esta información calcula las variables de grupo y en base a estas informaciones le muestra las tablas de los cálculos al participante. A partir de la primera ronda, el software:
 - Toma la información de la inversión de cada participante

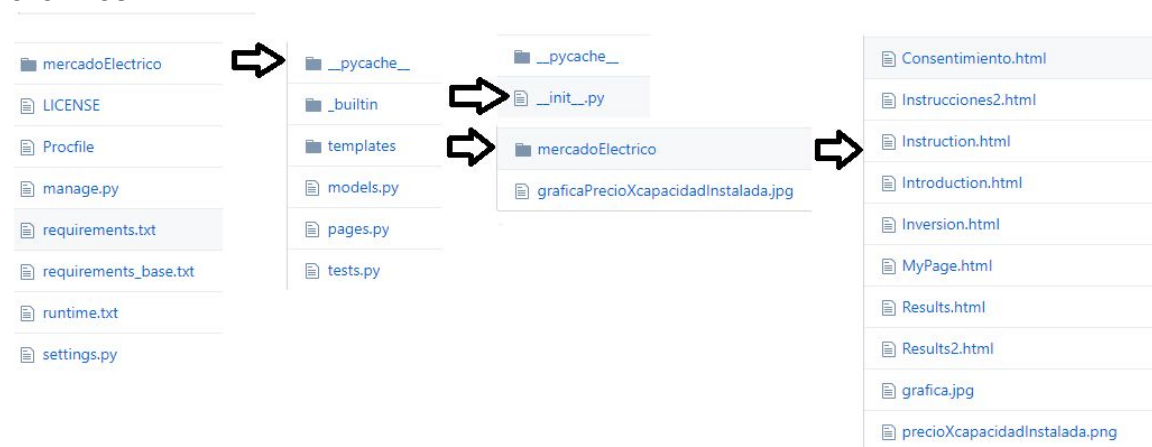
- actualiza las variables de capacidades en construcción e instaladas para cada año i
- Hace los cálculos para las capacidades instaladas y en construcción totales para cada participante
- Hace el cálculo del precio de la energía en el mercado, en función de la capacidad instalada total del mercado, el cálculo del margen de utilidad, en función del precio y el costo, y actualiza las variables estas variables de grupo
- Calcula la utilidad anual y total para cada participante
- Transforma toda la información de grupo y de cada participante para ser presentada en forma de tablas y gráficas
- El anterior proceso se repite n veces, siendo este el número de años (rondas) totales que dura el experimento.
- Al final, se hace el cálculo del incentivo a pagar al participante.

3. Implementación:

Esta herramienta gratuita y de fácil acceso se implementó gracias a la librería de código abierto oTree (<https://www.otree.org/>), y del lenguaje de programación Python. Esta librería se encuentra como plataforma con interfaz de usuario interactiva, como oTree Studio, de la plataforma oTree Hub. oTree se apoya a su vez de otra librería de desarrollo web de Python llamada Django. El despliegue del aplicativo se hace por medio de Heroku, la cual también es open source.

Arquitectura de la solución

Un proyecto de oTree puede albergar varias app o aplicaciones, la cual puede ser un experimento diferente. En este caso el proyecto solo alberga un experimento el cual es “Mercado Eléctrico de Energía Convencional”. A nivel de proyecto se maneja un archivo llamado requirements_base.txt donde se ponen los paquetes que necesita importar para funcionar la base del proyecto web, y también un archivo llamado requirements.txt, donde se especifican las diferentes librerías necesarias por las aplicaciones para funcionar. Por último, existe a nivel de proyecto un archivo llamado settings.py, donde se configuran parámetros generales del funcionamiento del proyecto. El directorio del proyecto está compuesto por las siguientes carpetas y archivos:



Los archivos referentes al experimento se encuentran dentro de la carpeta mercadoElectrico, donde se encuentran tres archivos y tres carpetas:

- Models.py: contiene las estructuras de datos que conforman a cada entidad del sistema, sus atributos y los métodos que implementan.
- Pages.py: define las paginas que conforman cada una de las vistas del software; con los métodos que definen su comportamiento y procesamiento de datos; secuencia que deben seguir las páginas del aplicativo.
- Test.py: Aquí se pueden desarrollar pruebas automatizadas al código, generando partidas con bots con parámetros predefinidos. Esta parte no se implementa en el proyecto.
- La carpeta templates contiene una imagen: graficaPrecioXcapacidadInstalada.jpg,
 - Dentro se encuentra otra carpeta llamada mercadoElectrico. Aquí se encuentran todos los archivos templates (html) que se renderizan y conforman las interfaces de usuario.
- La carpeta builtin contiene el archivo __init__.py que inicializa el experimento y todas sus clases.
- Las carpetas __pycache__ contiene archivos de la caché del software.

Desarrollo

En primer lugar se definió los atributos y métodos propios de cada entidad o clase del sistema. Esto se realizó en el archivo de models.py donde se implementaron las restricciones necesarias para cumplir con el análisis y el diseño. El archivo pages.py se encarga de definir como es la dinámica de las páginas dentro de la ejecución normal de la aplicación. Por cada pagina incluida en la secuencia del archivo pages.py debe existir un archivo .html en la carpeta de templates, los cuales combinan sintaxis html y comandos propios de oTree. Para mas detalle referirse al manual situado en el [repositorio del proyecto](#).

Despliegue

El despliegue del proyecto se realizó por medio de la plataforma Heroku, por medio de [oTree Hub](#). En el siguiente link se puede consultar la aplicación desplegada en la nube de Heroku: <https://mercados-electricos.herokuapp.com/demo/>

4. Validación:

El modelo teórico fue validado siguiendo la literatura de referencia para este proceso, validando tanto la estructura como el comportamiento. Primero se validó la estructura realizando pruebas como:

- Pruebas directas:
 - Pruebas de evaluación de estructura: Permiten verificar la consistencia de la estructura del modelo con el sistema real. Para esto se recurre al conocimiento de expertos, y a las teorías planteadas sobre el problema.
 - Prueba de evaluación de parámetros: Se verifica que todos los parámetros correspondan a elementos del sistema real, y que sus

valores numéricos concuerden con los valores observados o tengan lógica.

- Prueba de condiciones extremas directa: Esta prueba se realiza a las ecuaciones del modelo de manera manual, y sirve para encontrar fallas en la estructura del modelo, como para revelar variables omitidas. Para esto se deben considerar los máximos y mínimos imaginarios.
- Prueba de límites adecuados: Aquí se debe verificar si el modelo contiene las relaciones estructurales adecuadas para satisfacer el propósito del modelo, y si el nivel de agregación es el apropiado para el propósito del modelo.
- Prueba de consistencia dimensional: Para realizar esta prueba se debe evaluar la consistencia de las unidades de cada una de las variables del modelo en busca de verificar que sea consistente dimensionalmente.
- Pruebas indirectas
 - Prueba de condiciones extremas indirecta: En este caso se deben asignar valores extremos a algunos parámetros. Una vez hecho esto, el comportamiento del modelo se compara al comportamiento esperado del sistema real bajo estas mismas condiciones extremas.
 - Prueba de error de integración: Por medio de esta prueba se verifica si los resultados de la simulación son sensibles a la selección del paso del tiempo y el método de integración.

Seguido a la validación de la estructura se realizó la validación del comportamiento del modelo con:

- Prueba de reproducción del comportamiento: Examina qué tanto el comportamiento generado por el modelo concuerda con el comportamiento observado del sistema real. Cuando los datos de las series de tiempo históricas del sistema real se encuentran disponibles, el modelo debe ser capaz de reproducir los datos de forma similar.
- Análisis de sensibilidad: El objetivo del análisis de sensibilidad es determinar si se perciben cambios significativos en los datos numéricos o en el comportamiento del sistema cuando se realizan pequeñas variaciones en los valores iniciales del modelo (condiciones iniciales o parámetros).

La validación de la herramienta experimental fue transversal a los tres etapas descritas anteriormente (análisis, diseño e implementación) con el fin de verificar constantemente su funcionamiento correcto. Esta fue realizada por el equipo de desarrollo, por medio de pruebas unitarias (no automatizadas) a los módulos. Una vez completado este proceso de validación interno, se procedió a someter el software a prueba en una jornada de experimentos piloto con estudiantes de la Universidad Nacional de Colombia, sede Medellín.