

# EXPERIMENTOS ECONÓMICOS EN MERCADOS ELÉCTRICOS

---

MOME - MODELLING ELECTRICITY  
MARKETS - A SUPPLY GAME

**Software de Uso Libre**

Instituto Tecnológico Metropolitano -ITM  
Universidad Nacional de Colombia

Medellín, Antioquia, Colombia



Autores

Santiago Arango-Aramburo

Karla C. Álvarez-Uribe

Sebastián Bernal García

*Instituto Tecnológico Metropolitano - ITM*

*Universidad Nacional de Colombia –Sede Medellín*

Medellín, Antioquia, Colombia

2020

**Contenido**

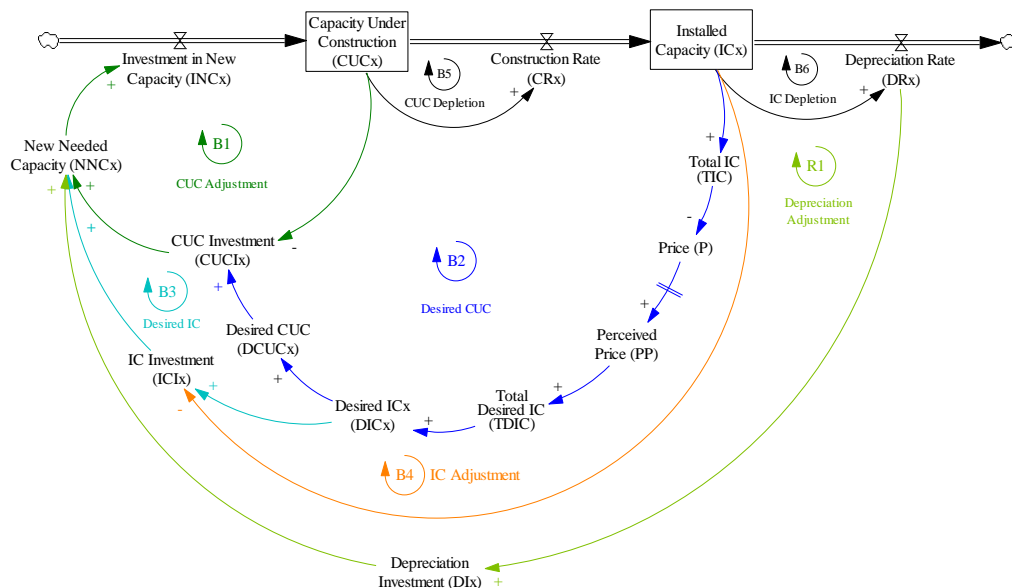
Introducción .....	3
Manual de Usuario .....	5
Preparación del Experimento .....	5
Desarrollo del experimento .....	11
Descarga de Datos .....	13
Pago a Participantes .....	14
Descripción del Programa .....	16
Arquitectura .....	16
Bienvenida .....	19
Consentimiento Informado .....	19
Instrucciones .....	20
Decisiones .....	22
Calculo de Resultados .....	24

## Introducción

A continuación, se encuentra el manual de usuario y la descripción del software de uso libre desarrollado. Este modelo se implementó primero usando dinámica de sistemas, un enfoque de modelación por computador que permite manejar sistemas con altos niveles de complejidad dinámica. Una vez validado este modelo matemático se procedió a programar el experimento haciendo uso de oTree.

El Software “MOME – Modelling electricity markets - a supply game” surge a partir de un modelo teórico sobre un mercado genérico de energía eléctrica bajo 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 – Modelling electricity markets - a supply game, se obtuvo gracias a los estudios encontrados en la literatura sobre: mercados de commodities, modelo cobweb, 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), 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.

Este modelo se basa en un modelo genérico de administración de un inventario. La estructura del modelo se presenta en la siguiente figura dónde se identifican 6 ciclos de realimentación negativos (B1, B2, B3, B4, B5 y B6), y un ciclo de realimentación positivo (R1). Además, se distinguen dos variables de estado o niveles (dentro de un rectángulo), las cuales representan la Capacidad en Construcción (CUC) y la Capacidad Instalada (IC).



El subíndice  $x$  denota el tipo de fuente de energía. El signo en la cabeza de cada flecha describe el tipo de relación entre dos variables así:  $x \rightarrow^+ y$  significa que  $\partial y / \partial x > 0$ ; y  $x \rightarrow^- y$  significa que  $\partial y / \partial x < 0$ . Todas las constantes se omitieron por claridad. Elaboración propia en Vensim (Ventana Systems Inc., 2013).

Para realizar este propósito se tuvieron en cuenta 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

(5 compañías) los cuales durante 70 rondas deben tomar decisiones para generar utilidades traducibles en dinero en el mundo real.

#### Consideraciones

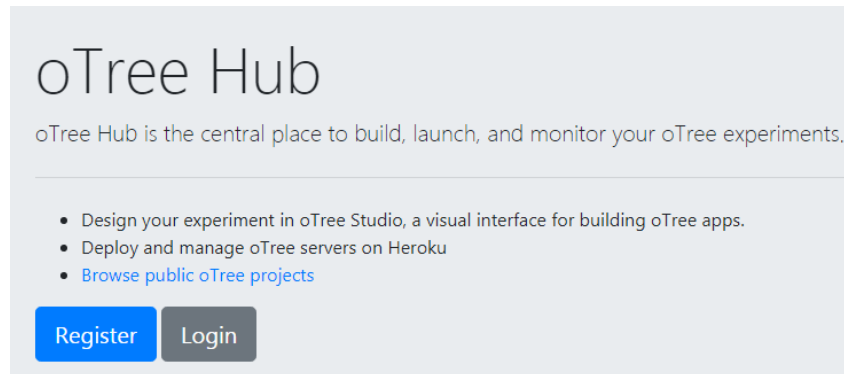
- Cada año (ronda) el participante (Compañía) tendrá que tomar decisiones de Inversión [unidades/año] en nuevas plantas, con el objetivo de maximizar la Utilidad Acumulada [\$] durante todo el experimento.
- La Utilidad Anual [\$] puede ser positiva o negativa (dependiendo del margen de utilidad, que es la diferencia entre el precio y costo de la energía).
- La Utilidad Acumulada [\$] es la suma de las Utilidades Anuales durante todo el juego y pueden 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. 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 tanto, la cantidad mínima que se puede invertir es 0 [unidades].
- La suma de la capacidad instalada y construida por cada participante no puede superar las 20 unidades, por tanto, la inversión no puede ser tal que se viole esta cantidad.
- El precio de la energía está dado en función de la capacidad total instalada del mercado. 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 e igual a 1 [\$/unidad].
- Todos los participantes inician con las mismas condiciones (5 compañías en total componen un mercado). 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).

## Manual de Usuario

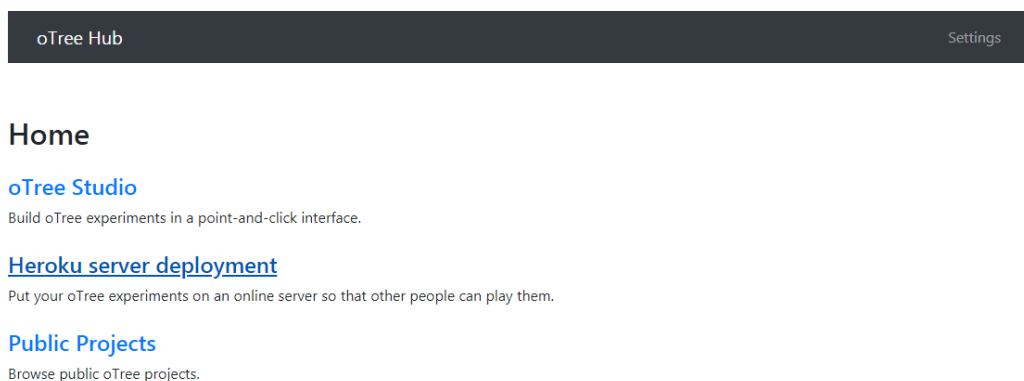
A continuación, se describen los procesos para preparar, y desarrollar el experimento, así como la definición del pago para los participantes y la descarga de los datos.

### Preparación del Experimento

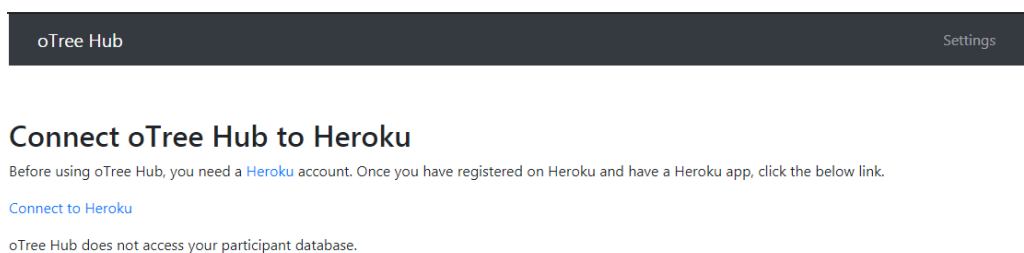
1. La mejor forma de desplegar un servidor para el experimento desarrollado en oTree, es hacer uso de oTree Hub ([www.otreehub.com](http://www.otreehub.com)).



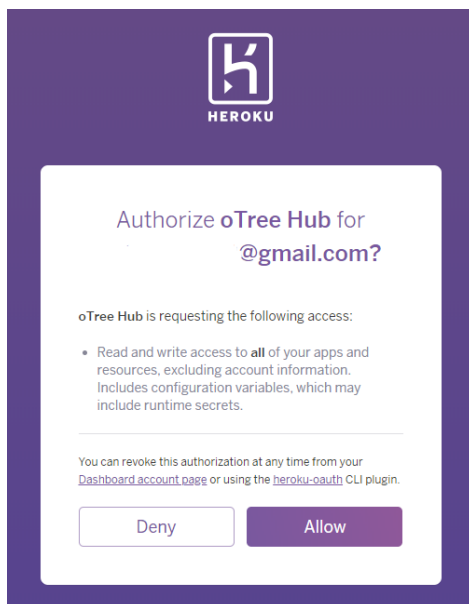
Una vez dentro de oTree Hub, se debe seleccionar la opción de desplegar un servidor de Heroku como se muestra en la siguiente imagen.



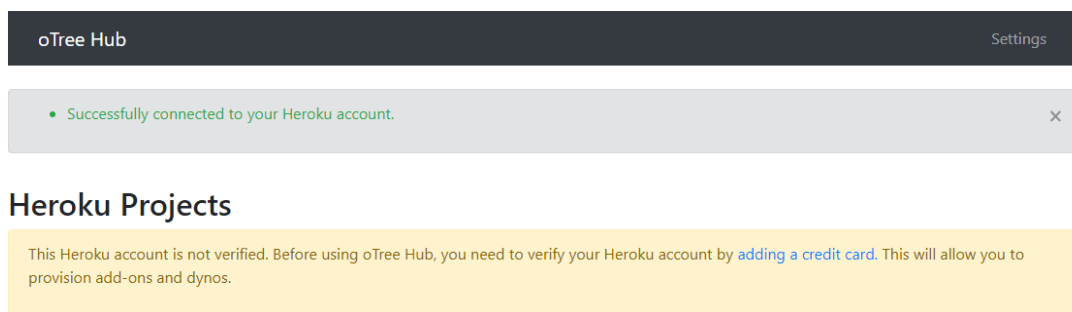
Heroku es un proveedor comercial de despliegue de aplicaciones en la nube, por esta razón, se debe tener una cuenta de Heroku.



2. Para crear la cuenta de Heroku, se debe tener en cuenta usar la misma cuenta de correo con la que se creó la cuenta de oTree Hub.



El plan gratuito de Heroku es suficiente para probar la aplicación, pero una vez se desee realizar un estudio, hay que cambiar a un servidor pago, el cual puede manejar más tráfico. Sin embargo, Heroku es bastante económico, ya que solo se paga por el tiempo real de uso. Si un experimento solo se hace un día, solo se paga 1/30 del costo mensual, lo que significa que se pueden realizar experimentos con poco dinero. Por esta razón, se debe agregar una tarjeta de crédito con el fin de poder verificar la cuenta de Heroku como se muestra en la siguiente imagen.



Una vez agregada la tarjeta de crédito, se procede a crear una nueva aplicación, directamente desde Heroku. Cada aplicación de Heroku corresponde a un proyecto completo de oTree, el cual puede contener más de una aplicación de oTree.

oTree Hub Settings

## Heroku Projects

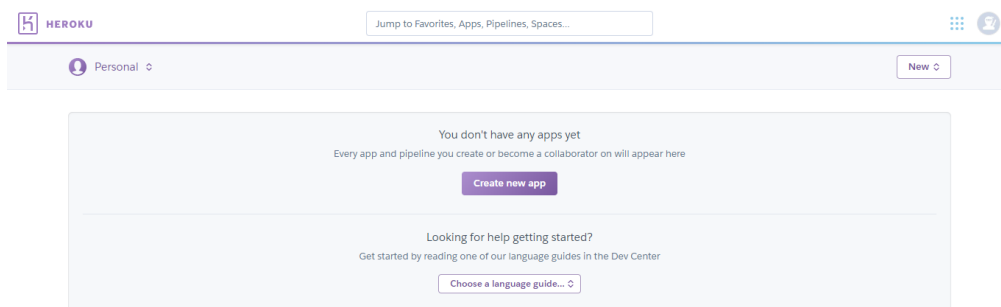
You don't have any active sites.

For a site to be listed here, you must be the owner, not a collaborator.

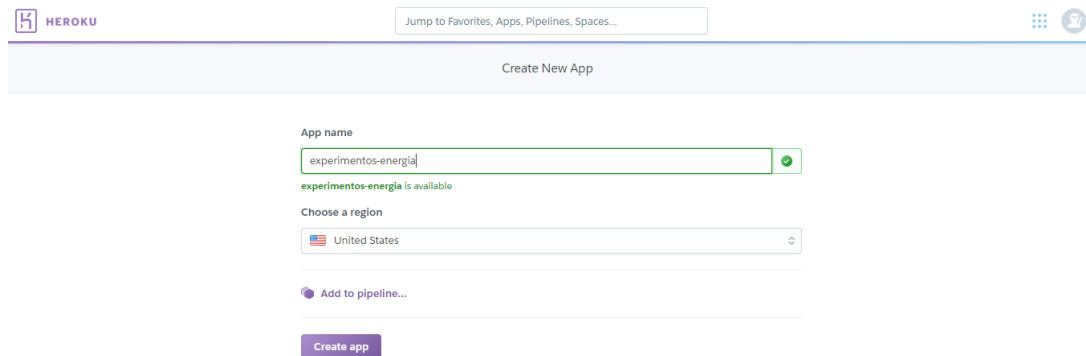
To create a Heroku site, log into [Heroku](#) and click the button to create a new app (Heroku calls its sites "apps", but each site corresponds to a whole oTree project, not a single oTree app). Then refresh this page.

After you finish a study, don't delete your site from Heroku, because oTree Hub's registration keys are not reusable. Instead, you should keep your site so that you can run your next study on it. You can deploy new code and even change the URL name. There is no need to create a new site.

3. Crear una nueva aplicación en Heroku es bastante sencillo. Solo se debe seleccionar la opción de crear una nueva app.

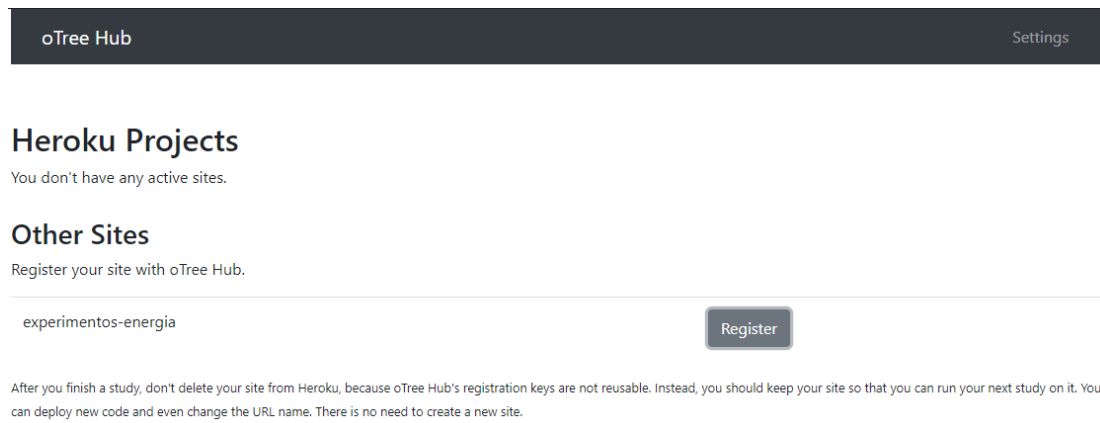


Para crear la aplicación lo único que se debe hacer es asignarle un nombre que no haya sido usado anteriormente en Heroku, y seleccionar la región, que para nuestro caso es Estados Unidos de América.

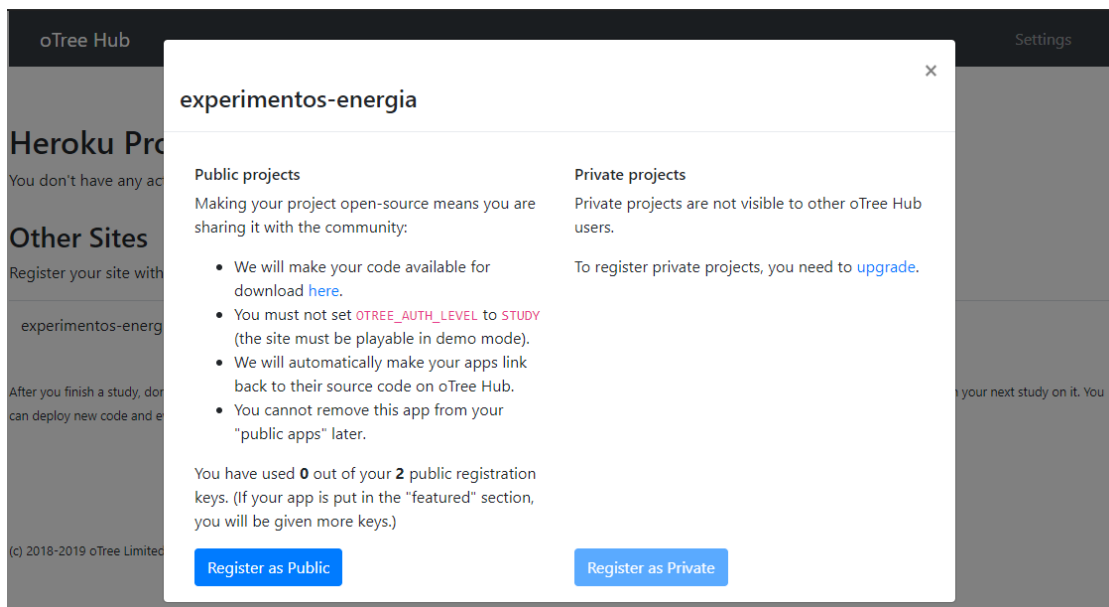


Una vez creada la aplicación, esta aparecerá automáticamente en oTree Hub. Esto se debe a que se las cuentas de Heroku y oTree Hub, fueron creadas con el misma cuenta de correo electrónico.





- Para empezar a hacer uso del sitio o app de Heroku en Otree Hub, este se debe registrar. Para esto hay dos opciones, registrarlo como público o registrarlo como privado. El registro público es gratuito e implica hacer el proyecto de código abierto. Esto significa compartir el código con la comunidad y permitir su descarga. Por el contrario, el registro privado no es gratuito y no permite que otros usuarios vean ni descarguen el código del software.



- Una vez registrado el proyecto, es necesario habilitar los <<Redis>> y los <<Postgres>> en la opción de Configuración. Allí, se debe seleccionar un plan para cada opción, dónde los planes gratuitos <<hobby-dev>> son solo para realizar pruebas. Por esta razón, si se desea realizar un estudio real se debe escoger otro plan pago, que cómo ya se mencionó antes es bastante económico ya que solo se paga por el tiempo real de uso.

oTree Hub Settings

Projects > experimentos-energia Deploy Configure Monitor

(Postgres and Redis are required for oTree to work. The rest of the settings are recommended.)

You have the Heroku hobby-dev Postgres plan, which is sufficient for development but not sufficient for a real study.  
 Note: changing your Postgres plan will create a new database. Any existing data will remain in the old database, but if you have any important data in your database, you should download it first. You will need to run `resetdb` to set up the new database.

Details about different Postgres plans [here](#)

Postgres plan:

- ☒ hobby-dev [just for testing]
- ☐ hobby-basic
- ☐ standard-0

You have the Heroku hobby-dev Redis plan, which is sufficient for development but not sufficient for a real study.  
 Details about different Redis plans [here](#)

Redis plan:

- ☒ hobby-dev [just for testing]
- ☐ premium-0
- ☐ premium-1

6. El último paso para hacer el despliegue consiste en subir un archivo con extensión `.otreezip`.

oTree Hub Settings

Projects > experimentos-energia Deploy Configure Monitor

First time setting up this project?

1. Go to the "Configure" tab and enable Redis
2. Upload your code below and wait for the build to succeed
3. Click the button below to reset your database.
4. Open your app at <https://experimentos-energia.herokuapp.com/>.

**Upload code to Heroku**

Upload the otreezip file you got from oTree Studio, or run `otree zip` to create an `*.otreezip`.

Seleccionar archivo Ningún archivo seleccionado Upload

Para crear un archivo con esta extensión, se deben copiar los siguientes archivos ubicados en la carpeta del proyecto de oTree:

- `requirements_base.txt`: Contiene las librerías base que necesita un proyecto de oTree.
- `requirements.txt`: Este llama a `requirements_base.txt` y además, contiene el nombre de otras librerías que el proyecto en específico necesita para su correcto desempeño.
- `settings.py`. Contiene configuraciones generales del proyecto cómo el idioma, el número de rondas, el tipo de moneda del payoff, configuraciones de la sesión, etc.

Estos tres archivos deben pegarse en la carpeta que se va a compilar para crear el archivo `.otreezip`. Esta carpeta, entonces, debe contener los tres archivos mencionados anteriormente, y una carpeta para cada aplicación que se deseen desplegar en la nube cómo se muestra a continuación:

Nombre	Fecha de modificación	Tipo	Tamaño
mercadoElectrico	5/11/2019 11:21 a. m.	Carpeta de archivos	
requirements.txt	27/09/2019 5:13 p. m.	Documento de te...	1 KB
requirements_base.txt	27/09/2019 5:13 p. m.	Documento de te...	1 KB
settings.py	5/11/2019 11:12 a. m.	JetBrains PyChar...	4 KB

En este caso, solo se va subir una aplicación que lleva por nombre **mercadoElectrico**, razón por la cual hay una carpeta con ese nombre. Dentro de esta carpeta se encuentran los archivos de Python, y una carpeta templates, con los archivos HTML.

Nombre	Fecha de modificación	Tipo	Tamaño
__pycache__	5/11/2019 11:21 a. m.	Carpeta de archivos	
_builtin	5/11/2019 11:21 a. m.	Carpeta de archivos	
static	5/11/2019 11:21 a. m.	Carpeta de archivos	
templates	5/11/2019 11:21 a. m.	Carpeta de archivos	
__init__.py	27/09/2019 5:27 p. m.	JetBrains PyChar...	0 KB
models.py	5/11/2019 11:09 a. m.	JetBrains PyChar...	15 KB
pages.py	29/10/2019 12:20 p. m.	JetBrains PyChar...	4 KB
tests.py	27/09/2019 5:27 p. m.	JetBrains PyChar...	1 KB

Cuando se tengan listos estos archivos y carpetas, desde la ventana de comandos (cmd) se ejecuta la orden <<otree zip>> dentro de la carpeta que contiene el archivo settings.py, para generar el archivo .otreezip.

Después de que el archivo .otreezip es subido con éxito se debe obtener un mensaje en esa misma página como el siguiente:

#### Build result

Your build succeeded.


- Así, el proyecto ya debe aparecer como activo en la página principal de oTree Hub. Para acceder a este, se debe hacer click en el icono que esta al final del nombre del proyecto, el cual es un link público.

oTree Hub Settings

## Heroku Projects

### Active Sites

▼

experimentos-energia 

Deploy

Configure

Monitor

Public link

For a site to be listed here, you must be the owner, not a collaborator.

After you finish a study, don't delete your site from Heroku, because oTree Hub's registration keys are not reusable. Instead, you should keep your site so that you can run your next study on it. You can deploy new code and even change the URL name. There is no need to create a new site.

Este link público permite correr el experimento (Demo), además de explorar y descargar el código fuente.

oTree Hub Settings

## experimentos-energia

### Links

- [Demo](#)
- [Browse source code](#)
- [Download](#)

You can set your name [here](#).

### Description

Bienvenido, seleccione el experimento en el que desea participar.  
To edit your app's description, go to `settings.py` and edit `DEMO_PAGE_INTRO_HTML`.

### Apps

- mercadoElectrico

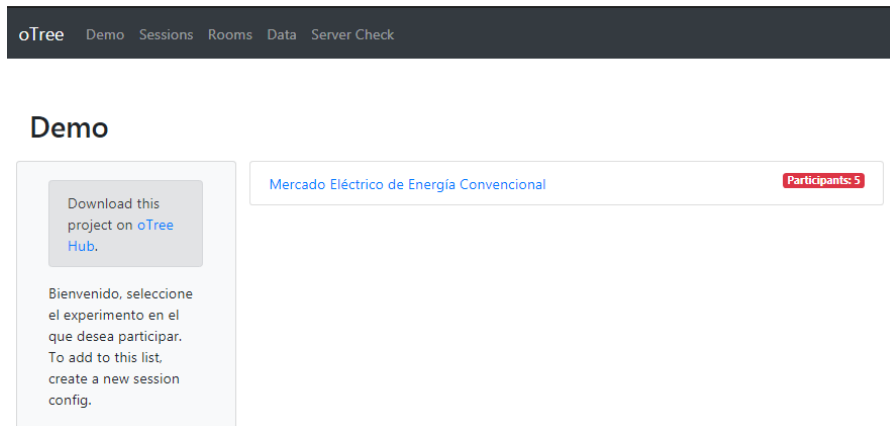
### Last updated

2019-10-25

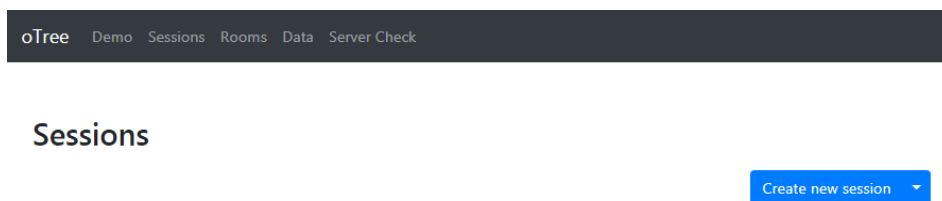
En caso de que se haya descargado el código, al final del link público aparece una nota con las instrucciones para descomprimir el archivo .otreezip

## Desarrollo del experimento

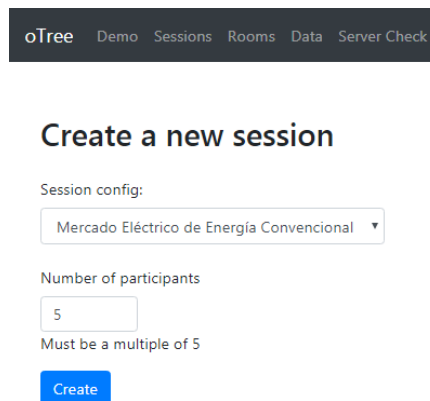
Para correr el experimento se debe ingresar al link de la aplicación creado en Heroku, que en este caso es: <https://mercados-electricos.herokuapp.com/demo/>. La página de inicio por defecto es Demo, la cual permite probar la aplicación pero no es recomendable para realizar estudios reales.



En el caso que se desee realizar un estudio real se debe crear una sesión.



Para esto se debe seleccionar la app y el número de participantes que van a tomar parte de la sesión.



Después de que se haya creado la sesión se generan los links <<sesión-wide>> o <<single-use>> para que los participantes puedan unirse al experimento. El <<sesión-wide>> link puede ser distribuido a los 5 participantes, sin embargo, no evita que la misma persona juegue dos veces, mientras que los <<single-use>> links tiene un código único para cada participante.

oTree Demo Sessions Rooms Data Server Check

## Mercado Eléctrico de Energía Convencional: session 7na9405a

[Description](#)
[Links](#)
[Edit](#)
[Monitor](#)
[Data](#)
[Payments](#)

You can either use the session-wide link, persistent links, or single-use links.

### Session-wide link

If it is impractical to distribute distinct URLs to each participant, you can give the following start URL to all 5 participants.

<https://mercados-electricos.herokuapp.com/join/271uv859b9/>

Note: unlike the other link modes, this does not prevent the same person from playing twice.

### Persistent links

If you want to give your participants permanent links that don't change, you should create your session in a [room](#).

### Single-use links

Below are single-use links, which you can distribute to your participants. Each link has a unique code for the participant.

P1	<a href="https://mercados-electricos.herokuapp.com/initializeParticipant/6yo4q60v">https://mercados-electricos.herokuapp.com/initializeParticipant/6yo4q60v</a>
P2	<a href="https://mercados-electricos.herokuapp.com/initializeParticipant/m36qjege">https://mercados-electricos.herokuapp.com/initializeParticipant/m36qjege</a>
P3	<a href="https://mercados-electricos.herokuapp.com/initializeParticipant/ttwaqgwo">https://mercados-electricos.herokuapp.com/initializeParticipant/ttwaqgwo</a>
P4	<a href="https://mercados-electricos.herokuapp.com/initializeParticipant/7aaqwhz6">https://mercados-electricos.herokuapp.com/initializeParticipant/7aaqwhz6</a>
P5	<a href="https://mercados-electricos.herokuapp.com/initializeParticipant/namggqqp">https://mercados-electricos.herokuapp.com/initializeParticipant/namggqqp</a>

Además, en esta misma interfaz puedo monitorear en tiempo real el avance de cada participante a través del experimento.

oTree Demo Sessions Rooms Data Server Check

## Mercado Eléctrico de Energía Convencional: session 7na9405a

[Description](#)
[Links](#)
[Edit](#)
[Monitor](#)
[Data](#)
[Payments](#)

ID in session	Code	Label	Page	App	Round	Page name	Status	Time on page
P1	6yo4q60v		14/350 pages	mercadoElectrico	3	Results2WaitPage	Waiting for P4, P5	<1 min ago
P2	m36qjege		14/350 pages	mercadoElectrico	3	Results2WaitPage	Waiting for P4, P5	<1 min ago
P3	ttwaqgwo		14/350 pages	mercadoElectrico	3	Results2WaitPage	Waiting for P4, P5	<1 min ago
P4	7aaqwhz6		14/350 pages	mercadoElectrico	3	Results2WaitPage	Playing	<1 min ago
P5	namggqqp		13/350 pages	mercadoElectrico	3	Inversion	Playing	<1 min ago

5/5 participants started.

[Advance slowest user\(s\)](#)

## Descarga de Datos

Los datos de la sesión se pueden descargar en formato .csv o .xls .

## Mercado Eléctrico de Energía Convencional: session 7na9405a

[Description](#)
[Links](#)
[Edit](#)
[Monitor](#)
[Data](#)
[Payments](#)

nd	actualRound	inversion	capacidad Instalada1	capacidad Instalada2	capacidad Instalada3	capacidad Instalada4	capacidad Instalada5	capacidad Instalada6	capacidad Instalada7
	1	0.0	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875
	1	0.0	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875
	1	0.0	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875
	1	0.0	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875
	1	0.0	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875	0.6875

You can download data in Excel or CSV format [here](#).

Además, también hay una documentación útil asociada a cada aplicación que detalla completamente las tablas SQL.

### Per-app

These files contain a row for each player in the given app. If there are multiple rounds, there will be multiple rows for the same participant. This format is useful if you are mainly interested in one app, or if you want to correlate data between rounds of the same app.

App	Data	Documentation
mercadoElectrico	<a href="#">Excel</a>   <a href="#">CSV</a>	<a href="#">TXT</a>

If the above links don't work for you, you can try the URLs `/ExportWide/` and `/ExportApp/your_app_name/` For xlsx, add `?xlsx=1` to the URL.

### Pago a Participantes

En la última pestaña de esta interfaz se encuentra el pago que debe recibir cada participante en tiempo real, teniendo en cuenta la definición que se hizo de este en el archivo *models.py* el cual se explica más adelante.

oTree Demo Sessions Rooms Data Server Check

## Mercado Eléctrico de Energía Convencional: session 7na9405a

[✈ Description](#)[🔗 Links](#)[✎ Edit](#)[👁 Monitor](#)[📄 Data](#)[💰 Payments](#)

Generated: 7 de Noviembre de 2019 a las 16:57

### Session

Session config	mercadoElectrico
Session code	7na9405a
Participation fee	15000,00 COP

### Participants

Code	Label	Progress	Wait page total	Payoff (bonus)	Total
6yo4q60v		15/350 pages	18 seconds	21263,00 COP	36263,00 COP
m36qjege		15/350 pages	13 seconds	21263,00 COP	36263,00 COP
ttwaqgwo		15/350 pages	8 seconds	21263,00 COP	36263,00 COP
7aaqwhz6		15/350 pages	3 seconds	21263,00 COP	36263,00 COP
namggqqp		15/350 pages	40 seconds	21263,00 COP	36263,00 COP



## Descripción del Programa

En esta sección se describe la arquitectura del software. Además, se presenta las páginas de Bienvenida, Consentimiento Informado, Instrucciones, Decisiones, y Calculo de Resultados que los participantes ven durante el desarrollo del experimento.

### Arquitectura

El proyecto está compuesto de forma más superficial de los siguientes archivos y carpetas:

- **Settings.py:** Tiene configuraciones básicas del proyecto y de las sesiones, como:
  - *REAL\_WORLD\_CURRENCY\_CODE:* Moneda del mundo real. En nuestro caso es COP (peso colombiano);
  - *real\_world\_currency\_per\_point:* El equivalente por punto ganado durante el experimento en dinero del mundo real. En nuestro caso igual a 1;
  - *participation\_fee:* Pago por participación. En nuestro caso 15.000 COP;
  - *name:* nombre de la app, que en nuestro caso es <<mercadoElectrico>>;
  - *display\_name:* Es el nombre que se muestra a los participantes, el cual es <<Mercado Eléctrico de Energía Convencional>> en nuestro caso.
  - *num\_demo\_participants:* Número de participantes en la sesión, que es diferente a los participantes por grupo, ya que una sesión puede tener varios grupos. En nuestro caso son 5 participantes.
  - *LANGUAGE\_CODE:* Es el código ISO-639 del lenguaje que se va usar en la app, que en nuestro caso es <<es>> (español).

Además, en este archivo también se pueden declarar las ROOMS, las cuales son sesiones que permanecen en el tiempo con links son permanentes para cada participante. Si se desea más información sobre esto, se puede consultar la documentación de oTree al respecto (<https://otree.readthedocs.io/en/latest/rooms.html>).

- *requirements\_base.txt:* Es un archivo en texto plano que contiene las librerías base para el proyecto en Python/Django, que al momento del desarrollo de este experimento es otree>=2.1.34.
- *requirements.txt:* Es un archivo en texto plano que contiene las librerías extra que necesita en específico nuestro proyecto en oTree. En este caso solo se hace necesario *psycopy2*=2.5.1. También, desde este archivo se llama a *requirements\_base.txt*, ya que el proyecto busca los requerimientos en el archivo *requirements.txt*

Los archivos anteriores son comúnmente utilizados en todos los proyectos de oTree. Además de estos, en la misma ubicación se encuentra la carpeta propia del proyecto, donde se encuentra toda la lógica de la aplicación (archivos .py pertenecientes al back-end) y los archivos del template, hojas de estilos (archivos .html, .css, y .js del front-end). Esta carpeta lleva por nombre mercadoElectrico, y contiene todos los archivos desarrollados para el proyecto, es decir, los .py que guían la lógica del proyecto y los .html que estructuran las páginas que el proyecto muestra en sus diferentes etapas (o fases) del flujo de trabajo. Es importante que esta carpeta, y los parámetros name y app\_sequence del archivo settings.py tengan el mismo nombre. Cabe resaltar que no son los únicos archivos que se pueden utilizar, pues también se pueden tener .js (javascript) para la reactividad de las paginas, hojas de estilo .css, y más. Dentro de esta carpeta se encuentran los siguientes archivos.

- *Models.py*: Aquí se definen los parámetros y métodos (o funciones) de cada una de las entidades (o clases) que conforman un proyecto de oTree<sup>1</sup>. Para este proyecto, este archivo está conformado por:
  - *class Constants(BaseConstants)*: Se crea una instancia de esta clase por cada sesión. Aquí se almacenan las constantes del proyecto, por ejemplo, el nombre de la url (igual a <<name>> del settings.py), número de jugadores por grupo (en nuestro caso 5), número de rondas (en nuestro caso 70). Además, para este proyecto en especial, como existen unas instrucciones que cada jugador debe conocer, y deben estar disponibles en cada ronda para este, definimos un parámetro *instructions\_template* que es la ruta relativa al html de las instrucciones.
  - *class Subsession(BaseSubsession)*: Es la sesión de cada grupo. Se definen la capacidad instalada final y el precio final.
  - *class Group(BaseGroup)*: Se crea una instancia de esta clase por cada grupo de participantes dentro de la sesión, la cual contiene las variables y métodos de grupo. Las variables definidas acá son el precio y la capacidad instalada total, los cuales cambian ronda a ronda, y el costo, el cual siempre es igual a uno. Entre los métodos están:
    - *calculosGrupo*: el cual se ejecuta al final de cada ronda para realizar los cálculos de capacidad instalada total, precio y los cálculos de cada participante (como las capacidades, ganancia y payoff) ronda a ronda.
    - *cálculosIniciales*: el cual permite rescatar el valor de las variables en la ronda inmediatamente anterior, ya que en cada ronda se reinician las variables, por lo que se debe mantener estas en el tiempo y poder cambiarlas ronda tras ronda.
  - *class Player(BasePlayer)*: Se crea una instancia de esta clase por cada participante que ingrese (con un link propio o aleatorio) a la sesión del proyecto. Algunas de las variables que se guardan son: el nombre, la cedula y su lugar de expedición. Se definen las variables del juego para cada participante: la inversión, las 4 capacidades en construcción y las 16 capacidades instaladas, también el total (valor en unidades) de las capacidades construidas e instaladas, el margen de utilidad (el cual se calcula en base a el costo y el precio), la utilidad (calculadas con el margen y las unidades instaladas), la utilidad acumulada (suma de la utilidad ronda a ronda), la capacidad instalada de los otros jugadores (suma de la capacidad instalada de cada uno de los otros jugadores). Esta clase tiene los siguientes métodos:
    - *calculosParticipante*: Acá se calcula la capacidad al final de la ronda, en base a las capacidades de la ronda anterior simulando que pasa una ronda de delay (retraso). Es decir, la inversión pasa a ser la capacidad en construcción del año 1, la capacidad en construcción del año 1 pasa a ser la capacidad en construcción del año 2 y así hasta que la capacidad en construcción del año 4, pasa a ser la capacidad instalada del año 1, la capacidad instalada del año 1 pasa a ser la capacidad instalada del año 2, y así hasta que la capacidad instalada del año 16 se deprecia totalmente y se olvida. Este método llama al siguiente método.
    - *calculosTotales*: Este toma las 4 capacidades en construcción y las 16 capacidades instaladas con el fin de obtener el total de capacidad instalada, y el total de capacidad en construcción.
    - *capacidadesPasadas*: es un método que se encarga de calcular las capacidades pasadas (ronda anterior) para mostrarlas en la ronda actual, como se dijo antes, esto es necesario porque oTree restaura todas las variables en cada ronda.

<sup>1</sup> Para más información sobre este archivo entrar a: <https://otree.readthedocs.io/en/latest/models.html>. Para una vista más conceptual de la estructura de models.py ingresar a: [https://otree.readthedocs.io/en/latest/conceptual\\_overview.html#conceptual-overview](https://otree.readthedocs.io/en/latest/conceptual_overview.html#conceptual-overview).

- *calculoGanancia*: se encarga de calcular el margen de utilidad, la utilidad y la utilidad acumulada de cada jugador; es llamada al final de cada ronda.
  - *calculoPayoff*: se encarga de calcular el payoff de cada participante como resultado de una decisión que se toma en base a su utilidad acumulada actual. Este se definió teniendo en cuenta que la utilidad acumulada de un participante promedio varía entre -300 y 300, siendo así:
    - utilidad acumulada  $\leq -300$  ----- payoff = 0, fee = 15.000
    - $-300 < \text{utilidad acumulada} < = 300$  ----- payoff =  $\text{abs}(-300 - \text{utilidad acumulada}) * 75$ ), fee = 15.000
    - utilidad acumulada  $> 300$  ----- payoff = 45.000, fee = 15.000
- *Pages.py*: Este archivo es el encargado de comunicar la información del archivo *models.py* con los templates HTML para la generación de formularios, gráficos y demás componentes dinámicos. Cada clase en este archivo equivale a una página que se muestra en el software:
  - *class Introduction(Page)*: Tiene un solo método *is\_displayed(self)* el cual se encarga de mostrar la página si y solo si el juego se encuentra en su primera ronda.
  - *class Consentimiento (Page)*: Se encarga de enviar al template los campos del formulario, en este caso de la información personal de usuario (nombre, cedula, y lugar de expedición). Al igual que *class Introduction*, tiene un método *is\_displayed(self)* que se encarga de que la pagina solo se muestre si el juego está en su primera ronda. Además tiene un método *def error\_message(self, values)* que toma los valores del formulario y realiza algunas validaciones como que el nombre y el lugar de la expedición de la cedula sean solo letras, y que el número de la cedula sea solo números y no sobrepase 12 dígitos. Cabe resaltar que en el proyecto entregado esta página esta desactivada, pero se puede activar fácilmente en la secuencia de páginas (*page\_sequence*), línea 110. En caso de activar esta página, es recomendable activar también *consentimientoWaitPage*, línea 111, la cual se encarga de generar una página de espera con la cual se retienen a los participantes que terminan de firmar el consentimiento, hasta que todos completen este, para darles a conocer las instrucciones.
  - *class Instrucciones2(Page)*: Esta se encarga de renderizar la página de instrucciones al principio del juego, y tiene también un método *is\_displayed(self)* que se encarga de que la pagina solo se muestre si el juego está en su primera ronda.
  - *class Inversion(Page)*: Se encarga de enviar al template el campo de inversión del formulario que se encuentra en *models.py*. Aquí, la función *is\_displayed(self)* se encarga de llamar a la función de grupo *calculosIniciales()* para mostrarle a cada participante su estado actual. El método *vars\_for\_template(self)* toma la información de algunas variables del *models.py* y la transforma en variables que se necesitan mostrar en el template. En nuestro caso se tomaron las variables precio, años (rondas), y capacidad instalada del mercado total de todas las rondas, y se almacenaron en listas que se pasaron por medio de un diccionario al template para renderizar en gráficos por medio de JavaScript. También tiene un método *error\_message(self, values)* que toma el dato del formulario de inversión y verifica que no se supere la capacidad total de un jugador (instalada total más construida total) en más de 20 unidades.
  - *class Results2WaitPage(WaitPage)*: Es la página de espera que se genera cuando un participante termina de hacer su inversión de la ronda, pues debe esperar que todos los participantes de su grupo terminen de tomar la decisión para realizar los cálculos de grupo (margen de utilidad, precio, y capacidad instalada total del mercado). Tiene un método *def after\_all\_players\_arrive(self)*, el cual se encarga que después de que todos los participantes lleguen a esta página (es decir todos hayan tomado la decisión de la inversión) se realicen los cálculos de grupo mencionados anteriormente, los cuales incluyen los cálculos para cada participante en específico.

- *class consentimientoWaitPage(WaitPage)*: Como se explicó anteriormente, esta página se encarga de retener a los participantes que terminen el consentimiento informado, hasta que todos terminen de completarlo. Tiene un método *is\_displayed(self)* que se encarga de que esta solo se renderice si el juego se encuentra en la primera ronda, como pasa con la página del consentimiento informado.
- *class Results2(Page)*: Es una página intermedia, para realizar los cálculos finales de la ronda y dar paso a la página de inversión del siguiente año, la cual mostrara la información actual del estado de cada participante y del mercado.
- Por último, se encuentra una lista llamada *page\_sequence* que tiene en orden las páginas que se renderizan a lo largo de una sesión del proyecto. Como se mencionó antes, se entrega con el consentimiento informado y su respectiva página de espera desactivadas.
- Test.py: Aquí se coloca la lógica de los robots para que desarrollen un juego. Esta parte no se programó.
- En la carpeta Templates/mercadoElectrico se encuentran los archivos html que se renderizan para estructurar la página a mostrar al usuario del software. Contiene también estilos css y lógica JavaScript (en inversión.html) para realizar gráficos interactivos.

## Bienvenida

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 comuniqué 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. La sesión se realizará en las siguientes partes: (1) Aceptación del consentimiento informado, (2) Lectura de instrucciones, (3) Experimento, (4) Liquidación y pago. El monitor indicará cuando empieza y cuando termina cada parte, y hará las aclaraciones necesarias. Por favor preste mucha atención y levante la mano si tiene alguna inquietud o inconveniente; el monitor se acercará y le resolverá las dudas.

## Consentimiento Informado<sup>2</sup>

Usted ha sido invitado a participar en este ejercicio que hace parte de un estudio sobre mercados eléctricos. Gracias a su participación en el juego y en las publicaciones que surjan de este, el sector científico se beneficiará de este estudio. Usted recibirá un pago en efectivo que depende de su desempeño en el juego. Al final del experimento deberá contestar algunas preguntas acerca del ejercicio y demográficas. Tanto la entrega del pago como la información sobre la forma como jugó cada participante en el juego se mantendrá en privado y para fines puramente académicos. Su participación en este ejercicio es totalmente voluntaria. Usted puede retirarse en cualquier momento. Sin embargo, si decide retirarse, no podrá recibir su pago pues es necesario jugar todos los periodos completos para poder calcular el pago final.

<sup>2</sup> Esta página se programó, pero se encuentra desactivada. Para activarla fácilmente se debe hacer en la secuencia de páginas (*page\_sequence*), línea 110. En caso de activar esta página, es recomendable activar también *consentimientoWaitPage*, línea 111, la cual se encarga de generar una página de espera con la cual se retienen a los participantes que terminan de firmar el consentimiento, hasta que todos completen este, para darles a conocer las instrucciones.

**ACEPTACION:**

Al pasar a la siguiente sección seleccionando el botón <<Next>>, yo declaro que comprendo la información anterior y mis derechos y compromisos durante este ejercicio. También reconozco que puedo retirarme en cualquier momento del juego y renunciar a recibir mi pago al final. Así, como recibir una copia de este consentimiento en cualquier momento.

Nombre: \_\_\_\_\_ C.C: \_\_\_\_\_ de \_\_\_\_\_

**Instrucciones**

**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 [\$]** durante todo el experimento.

**Utilidades:** La **Utilidad Anual [\$]** puede ser positiva o negativa y se calcula de la siguiente manera:

$$\text{Utilidad Anual} = \text{Margen de utilidad} * \text{Capacidad Instalada Total}$$

Por otro lado, la **Utilidad Acumulada [\$]** es la suma de las **Utilidades Anuales** durante todo el juego por lo que pueden ser negativas (perdida) o positivas (ganancia):

$$\text{Utilidad Acumulada} = \text{Utilidad Año 1} + \text{Utilidad Año 2} + \dots + \text{Utilidad Año 70}$$

**Capacidad en Construcción:** Su **Capacidad en Construcción Total [unidades]** depende de las decisiones de **Inversión** que usted tomó en años anteriores, ya que cada planta tiene un **tiempo de construcción de 4 años**. Es decir, desde el momento en que usted decide **Invertir** en la construcción de una nueva planta y el momento en que esta empieza a producir pasan **4 años**, tiempo después del cual su **Capacidad en Construcción** pasa a ser **Capacidad Instalada**.

**Capacidad Instalada:**

Su **Capacidad Instalada Total [unidades]** actual entonces equivale al número de plantas que están produciendo electricidad en el momento, las cuales tienen un **tiempo de vida útil de 16 años**. Es decir, cada planta produce energía durante **16 años** y luego sale del sistema. Esta **Capacidad Instalada Total [unidades]** no puede ser negativa, por lo que lo **mínimo** que usted puede invertir es **0 [unidades]**.

La **Capacidad Instalada del Mercado [unidades]** es igual a la suma de las **Capacidades Instaladas Totales** de cada jugador así:

$$\begin{aligned} \text{Capacidad Instalada del Mercado} = & \text{Capacidad Instalada Total Jugador 1} + \text{Capacidad Instalada} \\ & \text{Total Jugador 2} + \text{Capacidad Instalada Total Jugador 3} + \text{Capacidad Instalada Total Jugador 4} + \\ & \text{Capacidad Instalada Jugador 5} \end{aligned}$$

Debe tener en cuenta que la suma de su **Capacidad Instalada Total** y su **Capacidad en Construcción Total** no puede ser mayor que **20 [unidades]**.

Margen de Utilidad:

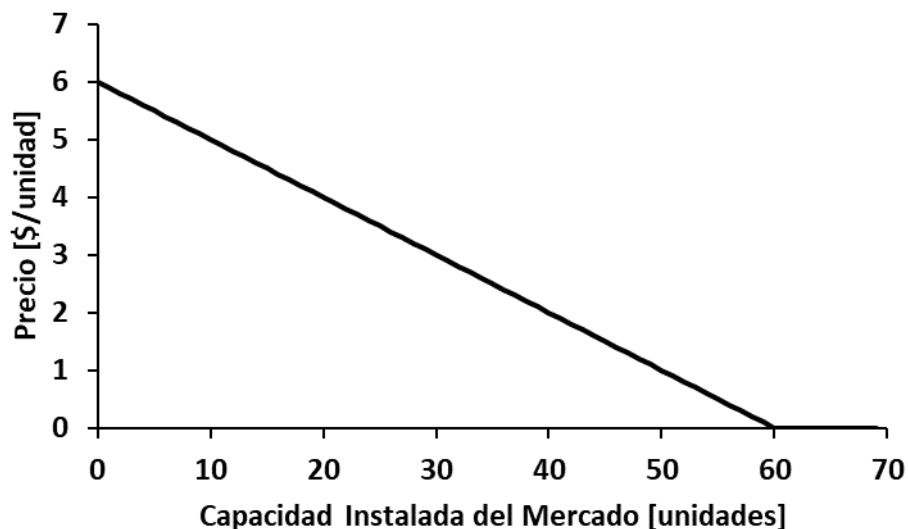
El **Margen de Utilidad** [\$/unidad] es igual a:

$$\text{Margen de utilidad} = (\text{Precio} - \text{Costo})$$

El **Costo** es constante y siempre es igual a **1 [\$/unidad]**.

Precio:

Por otro lado, el **Precio** [\$/unidad] del mercado depende de la **Capacidad Instalada del Mercado** como se muestra en la siguiente gráfica. Esta gráfica muestra una relación de proporcionalidad inversa entre el **Precio** y la **Capacidad Instalada del Mercado**. Esto significa que a mayor **Capacidad Instalada del Mercado** menor es el **Precio**, y que ha menor **Capacidad Instalada del Mercado** mayor el **Precio**.



Condiciones Iniciales: El total de las cinco compañías productoras de electricidad inician con las mismas condiciones. Durante todos los años anteriores, 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**). Por esta razón, las condiciones iniciales para cada variable son las siguientes:

**Capacidad en Construcción Año  $i$  = 0.6875 [unidades/año]**

**Capacidad en Construcción Total = 2.75 [unidades]**

**Capacidad Instalada Año  $i$  = 0.6875 [unidades/año]**

**Capacidad Instalada Total = 11 [unidades]**

**Capacidad Instalada del Mercado = 55 [unidades]**

**Precio = 0.5 [\$/unidad]**

**Costo = 1 [\$/unidad]**

**Margen de Utilidad = -0.5 [\$/unidad]**

**Utilidad Anual = -5.5 [\$]**

**Utilidad Acumulada = 0 [\$]**

### Decisiones

La siguiente gráfica muestra como los participantes deben tomar sus decisiones:

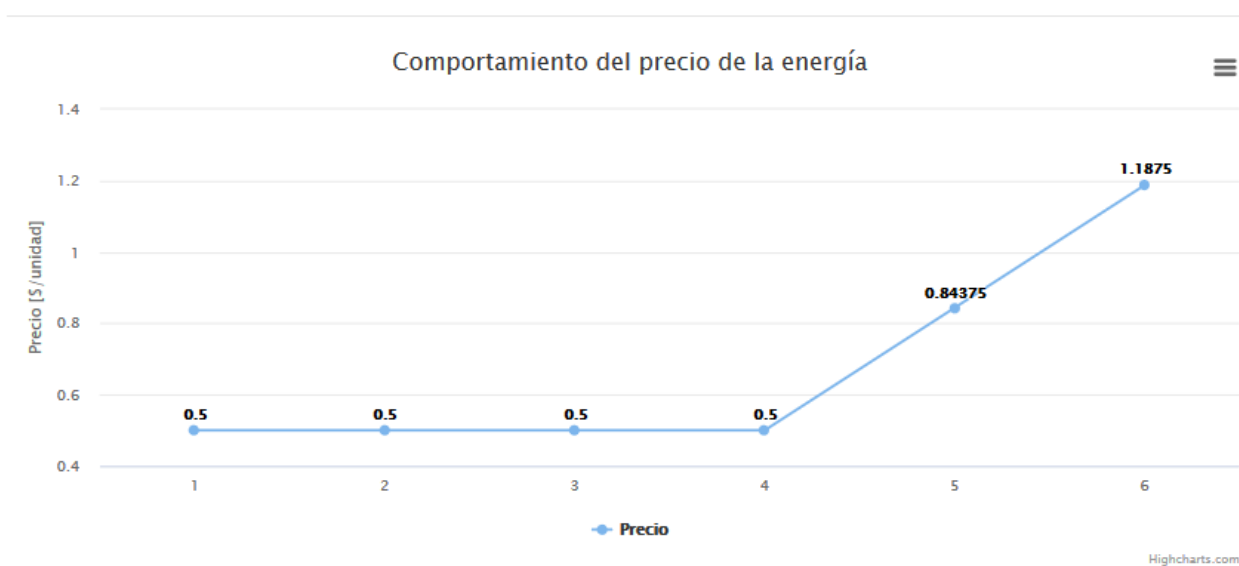
## Inversión

Eres el participante 5 y estas en el año 7. ¿Cuanto desea invertir en nueva capacidad este año?

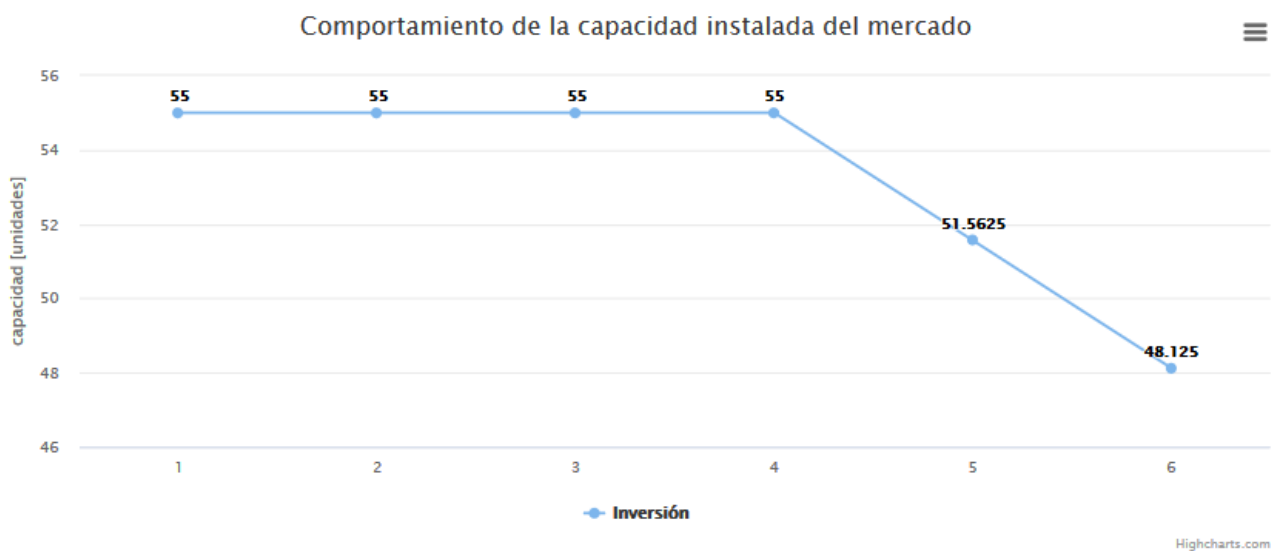
Inversion:

Siguiente

Además, debajo de este campo y en la misma página también se muestran los resultados de la ronda anterior.







Estado de la capacidad del participante 5 al final del año 6

Capacidad en Construcción [unidades]	
Capacidad en construcción año 1	0,0000
Capacidad en construcción año 2	0,0000
Capacidad en construcción año 3	0,0000
Capacidad en construcción año 4	0,0000
Capacidad en Construcción Total	0,0000

Capacidad Instalada [unidades]	
Capacidad instalada año 1	0,0000
Capacidad instalada año 2	0,0000
Capacidad instalada año 3	0,6875
Capacidad instalada año 4	0,6875
Capacidad instalada año 5	0,6875
Capacidad instalada año 6	0,6875
Capacidad instalada año 7	0,6875
Capacidad instalada año 8	0,6875
Capacidad instalada año 9	0,6875
Capacidad instalada año 10	0,6875
Capacidad instalada año 11	0,6875
Capacidad instalada año 12	0,6875
Capacidad instalada año 13	0,6875
Capacidad instalada año 14	0,6875
Capacidad instalada año 15	0,6875
Capacidad instalada año 16	0,6875
Capacidad Instalada Total	9,6250



## Información general histórica del participante 5 al final del año 6

Año	Capacidad Instalada Total	Capacidad Instalada Total de los otros Jugadores	Capacidad Instalada del Mercado	Precio	Costo	Margen de utilidad	Utilidad anual	Utilidad Acumulada
1	11,0000	44,0000	55,0000	0,5000	1,0000	-0,5000	-5,5000	-5,5000
2	11,0000	44,0000	55,0000	0,5000	1,0000	-0,5000	-5,5000	-11,0000
3	11,0000	44,0000	55,0000	0,5000	1,0000	-0,5000	-5,5000	-16,5000
4	11,0000	44,0000	55,0000	0,5000	1,0000	-0,5000	-5,5000	-22,0000
5	10,3125	41,2500	51,5625	0,8438	1,0000	-0,1563	-1,6113	-23,6113
6	9,6250	38,5000	48,1250	1,1875	1,0000	0,1875	1,8047	-21,8066

Adicionalmente, con el fin de facilitar una adecuada toma de decisiones por parte de los participantes también se muestran las instrucciones. Una vez, tomada la decisión el participante es dirigido a la siguiente página.

### Calculo de Resultados

Esta página se presenta una vez todos los participantes hayan tomado la decisión de inversión, y permite pasar a la siguiente ronda.

## Calculo de Resultados

Tus resultados ya fueron calculados.

Presiona el boton 'Siguiente' para visualizarlos y pasar a la siguiente ronda.

Siguiente