

Manual para implementar experimentos de oTree en Heroku

Universidad del Rosario

Noviembre de 2019

Elaborado por:

Alfredo Eleazar Orozco Quesada, Laura Prada Medina, Darwin Cortés y César Mantilla

Contenido

Introducción	2
1. Crear una cuenta en Heroku	2
2. Git.....	3
2.1. Instalar Git.....	3
3. CLI de Heroku.....	4
4. Proyecto de oTree.....	5
5. Transferir a un repositorio en Heroku (Git)	5
6. Ediciones finales	7
7. Adicionales.....	8
7.1. ¿Cómo actualizar la aplicación cuando esta ya se encuentra en Heroku?.....	8
7.2. Versión actualizada de oTree	9
7.3. Créditos.....	9

Ilustraciones

Ilustración 1. Sign up Heroku	2
Ilustración 2. Diagrama proceso.....	3
Ilustración 3. Terminal de comandos de Windows.....	4
Ilustración 4. Directorio en terminal de comandos	6
Ilustración 5. Interfaz log in App	8

Introducción

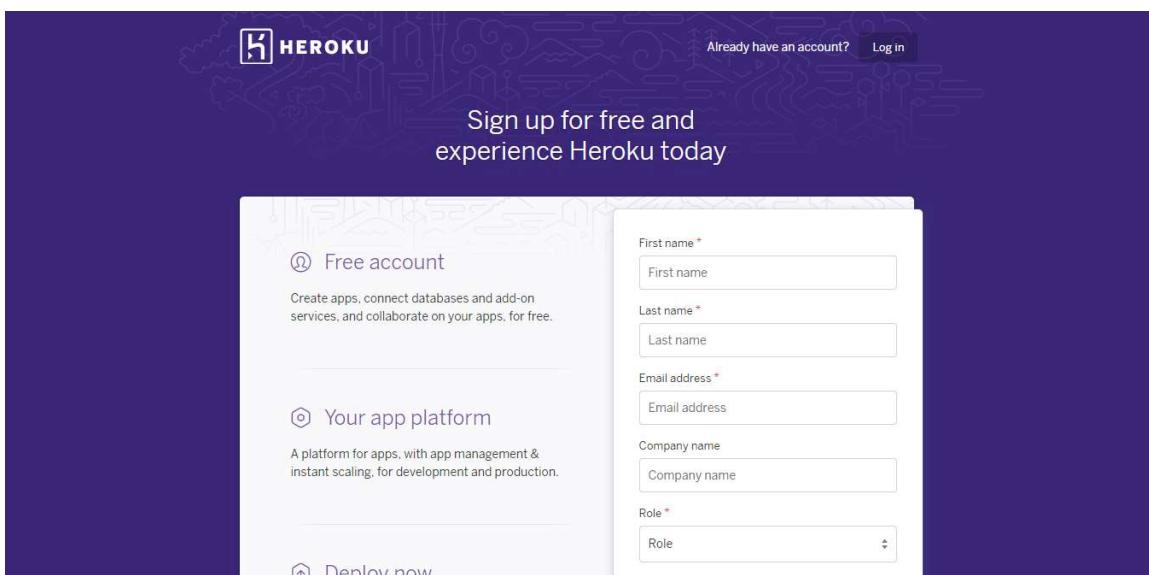
Todo experimento económico programado en oTree requiere el uso de un servidor para su implementación. Sin embargo, el montaje de servidores rigurosos y seguros no es un tema sencillo para personas ajena a la ingeniería de sistemas. Con el propósito de facilitar estas tareas, surgen las “Plataformas Como Servicio” (PaaS, en inglés) como Heroku, que permite ejecutar aplicaciones desde la nube, conectando clientes (jugadores) de manera simultánea desde diferentes computadores con acceso a internet. Así, esta es una herramienta de gran utilidad para la ejecución de experimentos económicos. En este manual se expone el procedimiento detallado para llevar un experimento desde un computador local hasta Heroku, para su implementación a través de internet. Además, se introducen algunos términos y herramientas necesarias durante el proceso de implementación.

1. Crear una cuenta en Heroku

En primer lugar, se debe ingresar al sitio de la empresa: <https://www.heroku.com/>. Heroku requiere una autenticación con correo electrónico. En el menú *Sign up* encontrara la opción para crear una cuenta de manera gratuita.

Heroku no cobra sus servicios básicos, pero tiene una amplia gama de extensiones que se pueden adquirir por un costo adicional. No obstante, la versión gratuita es suficiente para implementar la mayoría de los experimentos económicos. Si el investigador desea mayor poder de procesamiento, memoria, u otros, los servicios se cobran solo por el tiempo que se usan, permitiendo implementar temporadas completas de experimentación por un costo menor a la fracción de un mes. Con todo, de manera similar a abrir una cuenta de Facebook, para ingresar a Heroku solo es necesario llenar un formulario con datos básicos de contacto y confirmar la identidad por correo electrónico.

Ilustración 1. Sign up Heroku



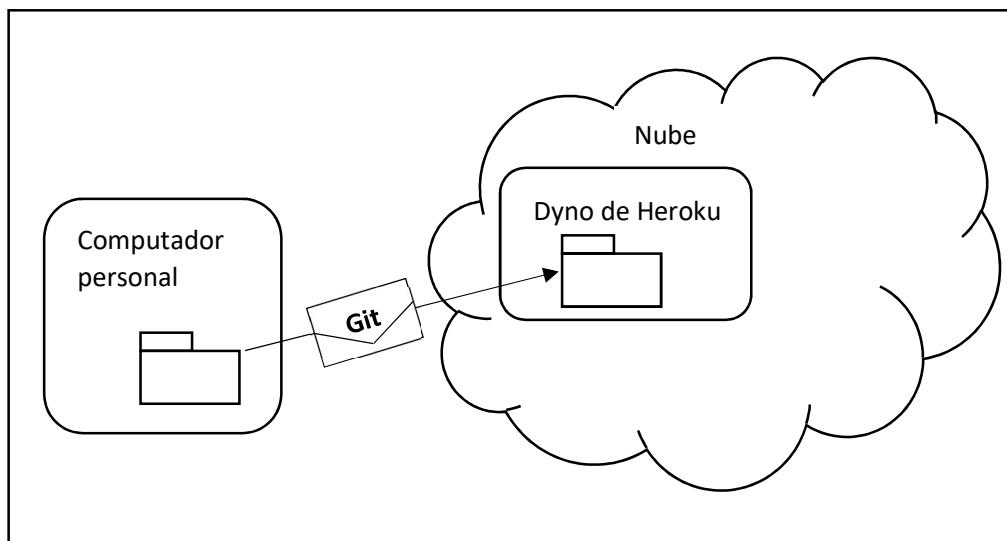
2. Git

Antes de comenzar, es importante entender la lógica de Heroku. Para trabajar con Heroku es necesario tener una carpeta con el proyecto (aplicación/experimento) en el computador personal. Una copia de la aplicación es transferida a *la nube*, donde los servidores en los que corren los procesos de Heroku prestan un “fragmento de computador” para que la aplicación sea ejecutada. A estos fragmentos de computador, Heroku los llama **Dynos**. Los **clientes** son personas que pueden acceder a las aplicaciones y solicitar su ejecución de manera remota a través de internet para ver e interactuar con los resultados. En nuestro caso, los clientes son tanto los sujetos experimentales, como el investigador que configura la sesión y supervisa el desarrollo del experimento.

Para transferir la aplicación de nuestro computador local a los Dynos de Heroku, es necesario tener un canal para enviarlos. Ese canal en este caso es un **manejador de versiones** llamado **Git**. Git tiene un paquete de instrucciones orientadas a editar proyectos de desarrollo de manera colaborativa. Así, los programas pueden ser editados en el computador de los desarrolladores para, cuando estén listos, ser enviados y que los usuarios puedan disfrutar sus versiones actualizadas. En este caso, solo vamos a usar Git para transferir cada versión del proyecto de nuestro computador local a los Dynos de Heroku.

El siguiente diagrama resume de manera simplificada la relación de todos los componentes.

Ilustración 2. Diagrama proceso



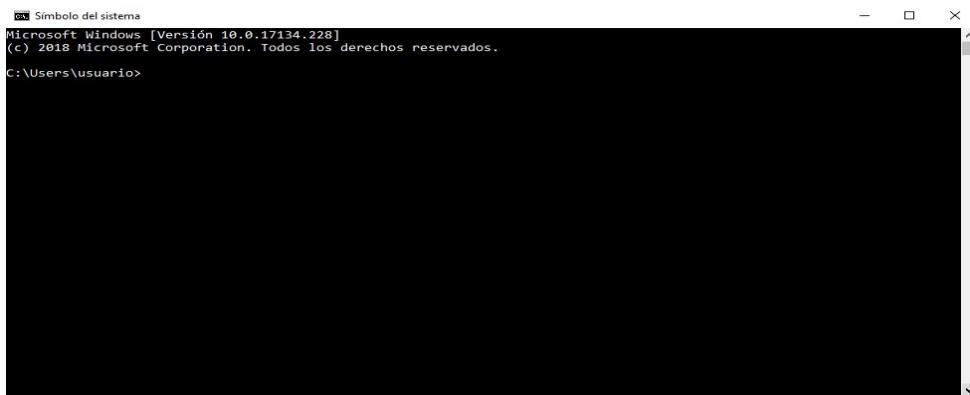
2.1. Instalar Git

Para instalar Git es necesario ir a la página de descargas oficial de Git <https://git-scm.com/downloads> y adquirir el instalador automático correspondiente al sistema operativo del equipo desde donde se subirá la aplicación.

Luego de instalado, no aparecerán accesos o íconos nuevos al programa pues Git se utiliza directamente desde la consola de comandos, como se explicará en esta sección (también se pueden instalar interfaces gráficas para usar Git mediante botones y diagramas. No obstante, en este caso no es necesario).

La primera vez que se usa Git es necesario configurar los datos de identidad, pues así, se puede registrar quién hizo cada modificación al programa y su correspondiente actualización. Para esto, hay que iniciar la terminal de comandos. Al buscar en las aplicaciones de Windows o de OS X la palabra “terminal”, la aplicación aparece de inmediato y se verá similar a la Figura 2:

Ilustración 3. Terminal de comandos de Windows.



Las instrucciones a continuación aplican igual para cualquier sistema operativo, dado que son instrucciones generales de Git. Es decir, una vez git está instalado en Windows, OS X, etc., sus comandos se pueden ejecutar universalmente.

Para establecer las variables de identificación debe ingresar a la terminal de comandos lo siguiente (una línea a la vez, seguida de la tecla enter):

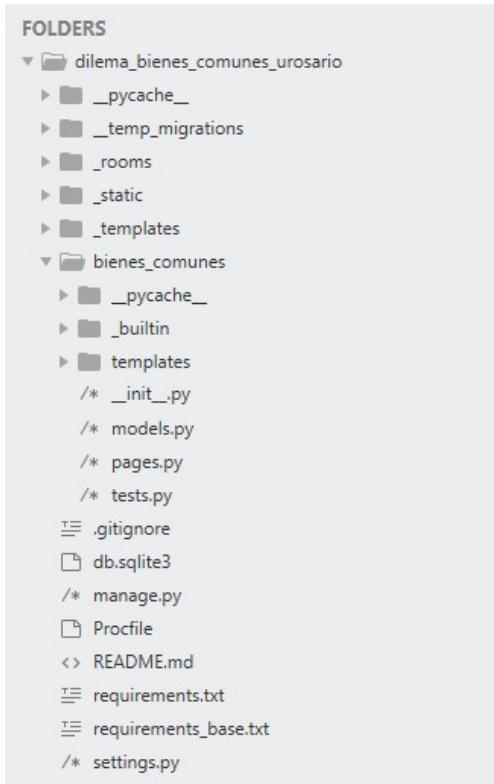
```
> git config --global user.name "John Doe"  
> git config --global user.email johndoe@example.com
```

Donde en lugar de John Doe y del correo example, se ubican sus datos personales.

3. CLI de Heroku

Una vez se tiene una cuenta en Heroku, es momento de prepararse para usarla. Lo primero que hay que hacer es instalar la Interfaz de Línea de Comandos (CLI) de Heroku. En términos simples, la CLI es un conjunto de programas que permiten al usuario darle instrucciones al computador relacionadas con procesos de Heroku. Estos comandos se ejecutan directamente desde la terminal de comandos, al igual que los comandos de Git. Accediendo a las descargas de Heroku CLI <https://devcenter.heroku.com/articles/heroku-cli> se encontrarán los instaladores automáticos para cada sistema operativo.

4. Proyecto de oTree



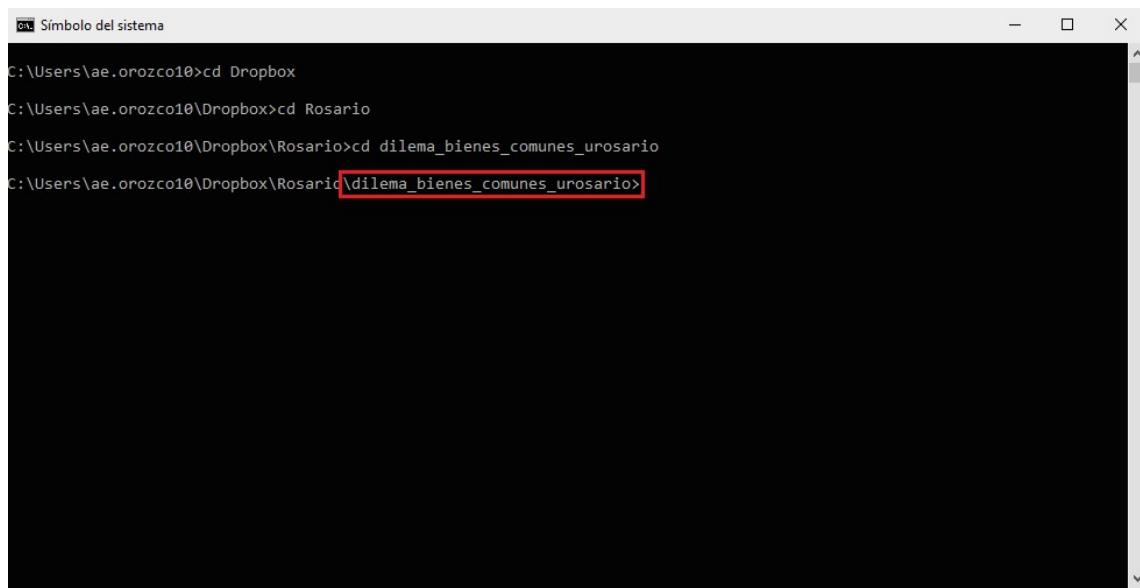
Hasta este momento se ha instalado Git (manejador de versiones para mandar el proyecto a los Dynos de Heroku) y la interfaz de comandos de Heroku (para darle instrucciones a Heroku relacionadas con el proyecto/experimento). Con estas dos herramientas se está listo para crear proyectos de oTree y enviarlos a Heroku.

Es importante que un proyecto de oTree tenga una estructura similar a la que se muestra en la figura a la izquierda, pues esta es la carpeta que se transfiere al Dyno de Heroku. En este caso, a la carpeta `dilema_bienes_comunes_urosario` se le llama carpeta raíz (root) del proyecto. En esta se encuentran todos los archivos y subdirectorios de la aplicación. Dado que oTree utiliza como base el lenguaje Python, hay varios archivos con extensión `.py`. Además, hay algunos archivos con información y configuración general como los `.txt` y el archivo `.md`. El archivo `Procfile` le indica a Heroku con qué procesos iniciar el servidor. El archivo `.gitignore` indica carpetas o archivos que no se envían al Dyno de Heroku al invocar el comando en git.

5. Transferir a un repositorio en Heroku (Git)

Tras crear la cuenta en Heroku, e instalar Git y Heroku CLI, se debe ejecutar Powershell. Una vez se abre la terminal de comandos se muestra un directorio, respecto al cual se ejecutan todos los comandos que se ordenen en la terminal. Para transferir la aplicación de oTree desde el computador local hasta Heroku se debe abrir la terminal y navegar con ella hasta ingresar a la carpeta raíz del proyecto. El comando `cd`, seguido del nombre de una carpeta permite ingresar a cada subcarpeta disponible. La siguiente figura muestra cómo debe verse la línea de comandos antes de comenzar a ejecutar instrucciones para transferir la aplicación.

Ilustración 4. Directorio en terminal de comandos



```
C:\ Símbolo del sistema
C:\Users\ae.orozco10>cd Dropbox
C:\Users\ae.orozco10\Dropbox>cd Rosario
C:\Users\ae.orozco10\Dropbox\Rosario>cd dilema_bienes_comunes_urosario
C:\Users\ae.orozco10\Dropbox\Rosario\dilema_bienes_comunes_urosario>
```

En caso de querer salir de una subcarpeta a su correspondiente carpeta contenedora, se puede ejecutar el comando **cd ..** que retrocederá un nivel en la estructura de carpetas.

Una vez allí, se hace inicio de sesión en Heroku con los datos ingresados al crear la cuenta, con el comando:

```
> heroku login
```

El programa le solicitará la información necesaria.

Posteriormente se inicia un repositorio local que se enviará a Heroku, para esto, se ejecuta:

```
> git init
```

Luego, se crea un proyecto en Heroku que albergará las configuraciones del servidor. Para esto, se emplea el comando:

```
> heroku create nombre-del-proyecto
```

Con esto, en Heroku aparecerá un proyecto con el nombre escogido. Además, la aplicación tendrá un enlace permanente de la forma **nombre-del-proyecto.herokuapp.com**.

A continuación, se debe incluir una funcionalidad adicional a la aplicación, llamada Redis. Para esto, se ejecuta:

```
> heroku addons:create heroku-redis:hobby-dev
```

Advertencia

oTree recomienda instalar Redis en su versión paga cuando se corra el estudio (*Premium 0*) porque admite más memoria y evita errores que puedan derivarse de esto. Sin embargo, esto generará un costo mensual de 15USD / mes (prorrateado a la fracción del mes que permanezca instalado). Por lo tanto, este manual recomienda prototipar con la versión gratuita (*hobby-dev*), y tal vez correr un piloto con un grupo de magnitud similar a la de las sesiones definitivas. Para instalar la versión gratuita, se debe ejecutar el comando anterior. No obstante, si usted decide usar la versión paga el comando que deberá usar es el siguiente:

```
> heroku addons:create heroku-redis:premium-0
```

A continuación, se prepara la aplicación para enviarla. Se ejecuta:

```
> git add .  
> git commit -am "Comentario sobre la versión que se manda"
```

Y finalmente, se envía a Heroku:

```
> git push heroku master
```

Adicional, se debe inicializar la base de datos de la aplicación con el comando:

```
> heroku run "otree resetdb"
```

Con todo lo anterior, el servidor habrá quedado montado en Heroku. Para acceder a la aplicación se puede ingresar al enlace nombre.herokuapp.com, o bien ejecutar

```
> heroku open
```

6. Ediciones finales

Antes de correr el experimento es necesario hacer dos configuraciones más: deshabilitar el modo de desarrollador “DEBUG”, y proteger por contraseña la aplicación. Para esto es necesario verificar que en el documento settings.py, en el directorio raíz, se haya definido un ADMIN_USERNAME, entre comillas así:

```
ADMIN_USERNAME = 'nombre_del_admin'
```

Por otro lado, se deben configurar desde la terminal de comandos un par de variables que van a quedar determinadas en Heroku, no en el computador local.

Primero, se deshabilita el estado de DEBUG asignando la variable de producción del experimento:

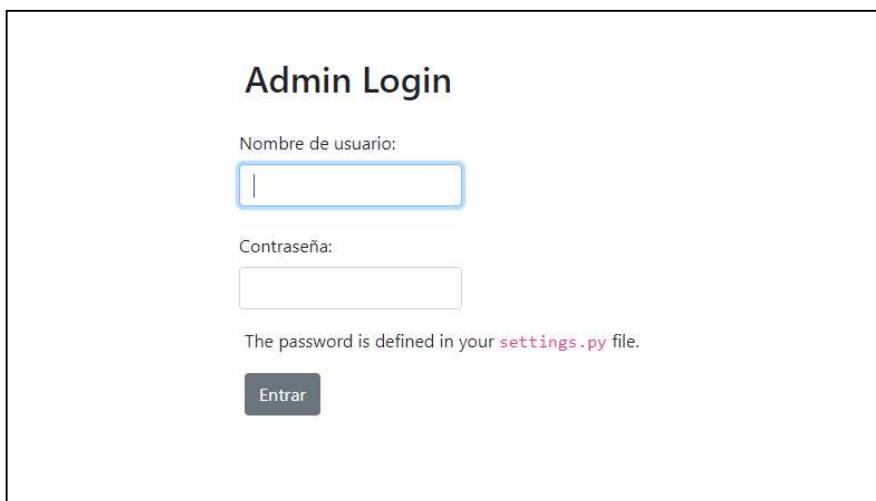
```
> heroku config:set OTREE_PRODUCTION=true
```

Posteriormente se configuran las variables de autenticación

```
> heroku config:set OTREE_AUTH_LEVEL=STUDY  
> heroku config:set OTREE_ADMIN_PASSWORD=contrasenaDeseada
```

De esta manera, la aplicación estará protegida por un usuario de administrador definido en el documento `settings.py`, y una contraseña, definida en las variables de entorno de Heroku. Luego de estas modificaciones al ingresar a la aplicación usted se encontrará con una pantalla como la de la figura 5.

Ilustración 5. Interfaz log in App



7. Adicionales

7.1. ¿Cómo actualizar la aplicación cuando esta ya se encuentra en Heroku?

Es usual que surja más de una modificación a las aplicaciones cuando estas ya se encuentran en el servidor (heroku en este caso). Para poder modificarlas, el procedimiento es bastante sencillo y se puede realizar desde cualquier equipo, pues no requiere que usted se encuentre en el computador desde donde envió la aplicación al servidor. No obstante, si es necesario que el equipo cuente con los requerimientos necesarios para mandar un archivo al servidor.

Habiendo dicho eso, el primer paso es el ingreso a heroku desde la terminal de su computador.

```
> heroku login
```

Una vez se ha autenticado usted ya podrá modificar las aplicaciones de su cuenta. Para ello, deberá generar una copia local de la aplicación disponible en el servidor.

```
> heroku git:clone -a nombre_aplicacion
```

```
> cd nombre_aplicacion
```

Donde nombre aplicación es el nombre de la aplicación en Heroku. Con una copia local de la aplicación usted puede realizar las modificaciones que necesite. Guarde los cambios e inicie de nuevo la transferencia de su aplicación a la nube.

```
> git add .  
> git commit -am "make it better"  
> git push heroku master
```

Con esto usted podrá actualizar su aplicación cuantas veces sea necesario.

7.2. Versión actualizada de oTree

Se recomienda, en caso de ser desarrollador de la aplicación, mantener una versión de oTree actualizada para cada nuevo proyecto. Para esto, se debe instalar Python y oTree. Para instalar debe descargar el instalador de la página oficial, y seguir las instrucciones de la página:

Una vez se tiene Python instalado, se emplea el instalador de este lenguaje para instalar oTree. En la ventana de comandos se ingresa la siguiente instrucción:

```
> pip3 install -U otree
```

Con esto, es posible verificar la versión actual de otree:

```
> otree --version
```

Esta versión debe reemplazarse en el documento **requirements_base.txt** del directorio raíz del proyecto:

otree>=X.X.X

Advertencia

Se recomienda no actualizar la versión de oTree durante la programación y ejecución de un proyecto, pues algunas de sus funcionalidades pueden cambiar.

7.3. Créditos

Finalmente, por requerimiento de licencia de oTree, al publicar investigación que emplee códigos programados con oTree, se debe citar el siguiente documento:

Chen, D.L., Schonger, M., Wickens, C., 2016. oTree - An open-source platform for laboratory, online and field experiments. Journal of Behavioral and Experimental Finance, vol 9: 88-97)