Técnicas de Simulación

Ing. Jorge Moya

Manual Técnico

Sánchez Cobeña Cristóbal Andrés

Octavo Nivel “A”

Manta, 11 de septiembre del 2017

Manual Técnico

Contenido

Manual técnico 3
Librerías y Versiones Utilizadas: 3
Descargas e instalación: 4
Python 2.7.13 4
Librerías: 4
Herramientas 5
CSS3 5
Javascript 6
Herramientas utilizadas y como usarlas 7
¿Cómo Iniciar este proyecto en su equipo con PyCharm? 8
Primeros pasos con HEROKU 14
¿Como subirlo a un Servidor (Heroku)? 15
¿Como modificar nuestro proyecto subido en Heroku? 19

Manual técnico

El presente manual tiene como finalidad describir la instalación y la ejecución del proyecto realizado para el sistema web de simulación, la siguiente aplicación web fue desarrollada con Python 2.7.13, la librería principal es Django.

Front-end: Se utilizó HTML:5, Librerías CSS3, Librerías JavaScript y plot.ly

Back-end: Python y Django

Librerías y Versiones Utilizadas:

- Python 2.7.13
  - Librerías de Python:
    - Django 1.11.2
    - Numpy 1.31.1
    - Scipy 0.19.1
    - PIP
    - whitenoise
- CSS 3
  - Librerías CSS
    - Bootstrap 4.0
    - ValidationEngine.jquery.css
    - Styles Personalizados
- JavaScript
  - Librerías JavaScript
    - Bootstrap 4.0
    - Bootstrap Validate
    - JQuery.validate
    - JQuery-3.2.1
    - Plotly-latest
    - validaciones
    - Funciones Personalizadas
- HTML 5
- Heroku CI (servidor Web)

Descargas e instalación:

Python 2.7.13

https://www.python.org/downloads/release/python-2713/

Librerías:

PIP:

Ejecutar en Consola: python get-pip.py

DJANGO: https://www.djangoproject.com/download/

Opcion1:

```
pip install Django==1.11.2
```

Opción 2:

```
git clone https://github.com/django/django.git
```

NUMPY:

Opcion1:

```
pip install numpy
```

Opcion2:

```
git clone https://github.com/numpy/numpy.git numpy
```

SCIPY:

Opcion1:

```
pip install scipy
```

Opcion2:

```
git clone https://github.com/scipy/scipy.git scipy
```

WHITENOISE:

```
pip install whitenoise
```

PIPENV:

```
pip install pipenv
```

Herramientas.

PYCHARM 2017.2 IDE

https://www.jetbrains.com/pycharm/download/#section=windows

HEROKU CLI

https://devcenter.heroku.com/articles/heroku-cli#download-and-install

CSS3

Bootstrap 4.0

http://getbootstrap.com/docs/4.0/getting-started/download/

TIPO DE LETRA

https://fonts.googleapis.com/css?family=Montserrat

Styles Personalizados

Los archivos de estilos (css) personalizados se encuentran en el proyecto:

* bootstrap.css
* bootstrap-theme.css
* estilo.css
* multi-step-form.css

* style.css

* validationEngine.jquery.css

Javascript

Bootstrap 4.0

http://getbootstrap.com/docs/4.0/getting-started/download/

JQuery-3.2.1

https://jquery.com/download/

Plotly-latest

https://cdn.plot.ly/plotly-latest.min.js

Funciones Personalizadas

Los archivos de JavaScript (.js) personalizados y externos se encuentran en el proyecto:

* Bootstrap Validate

* funciones.js

* jquery.validate.min.js

* jquery.validationEngine-2.6.2.js

* knockout-min.js

* multi-step-form.js

* npm.js

* valid.js

Herramientas utilizadas y como usarlas

PYCHARM 2017.2 IDE

Instrucciones de instalación

WINDOWS:

1. Ejecute el archivo pycharm-2017.2.3.exe que inicia el Asistente para la instalación.

2. Siga todos los pasos sugeridos por el asistente. Preste especial atención a las opciones de instalación correspondientes.

MAC OS:

1. Descargar el archivo de imagen de disco pycharm-2017.2.3 .dmg macOS

2. Montarlo como otro disco en su sistema.

3. Copie PyCharm en su carpeta de aplicaciones.

LINUX:

1. Copie el pycharm- 2017.2.3 .tar.gz en la ubicación de instalación deseada (asegúrese de tener permisos rw para ese directorio)

2. Descomprima el pycharm- 2017.2.3 .tar.gz utilizando el siguiente comando:

```
tar -xzf pycharm- 2017.2.3 .tar.gz
```

3. Quitar el pycharm- 2017.2.3 .tar.gz para ahorrar espacio en disco (opcional).

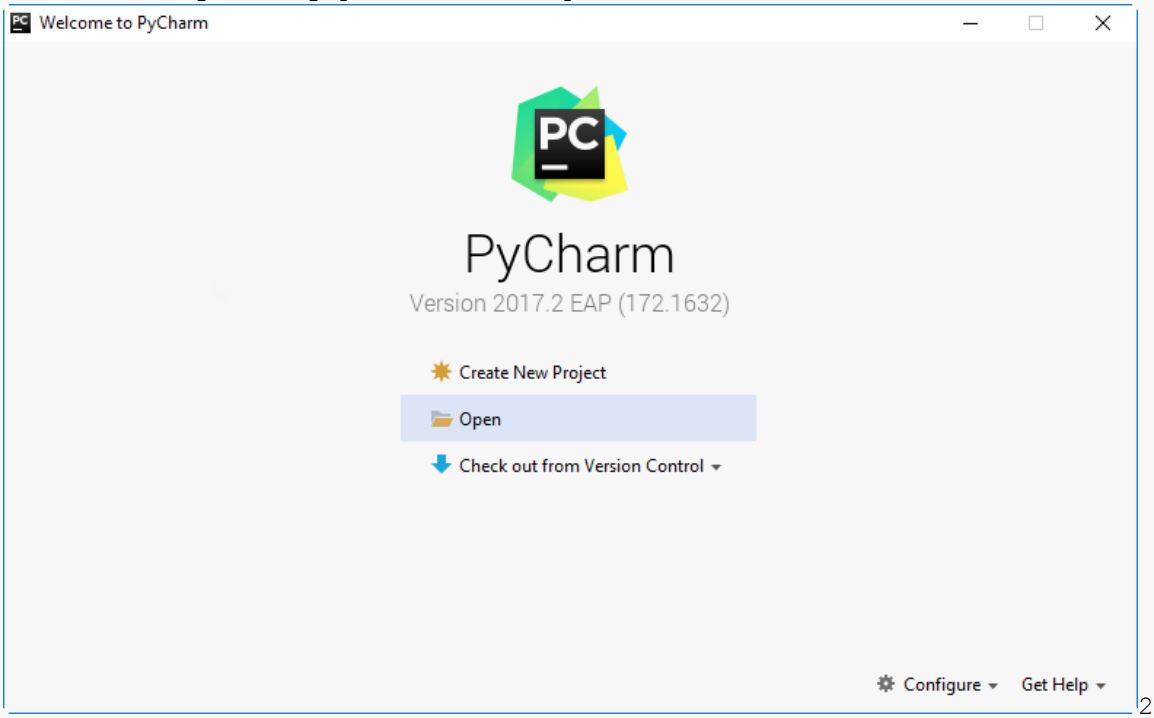
4. Ejecutar pycharm.sh desde el subdirectorio bin.

PARA REALIZAR UN PROYECTO POR FAVOR REVISAR LA DOCUMENTACION OFICIAL DE PyCharm

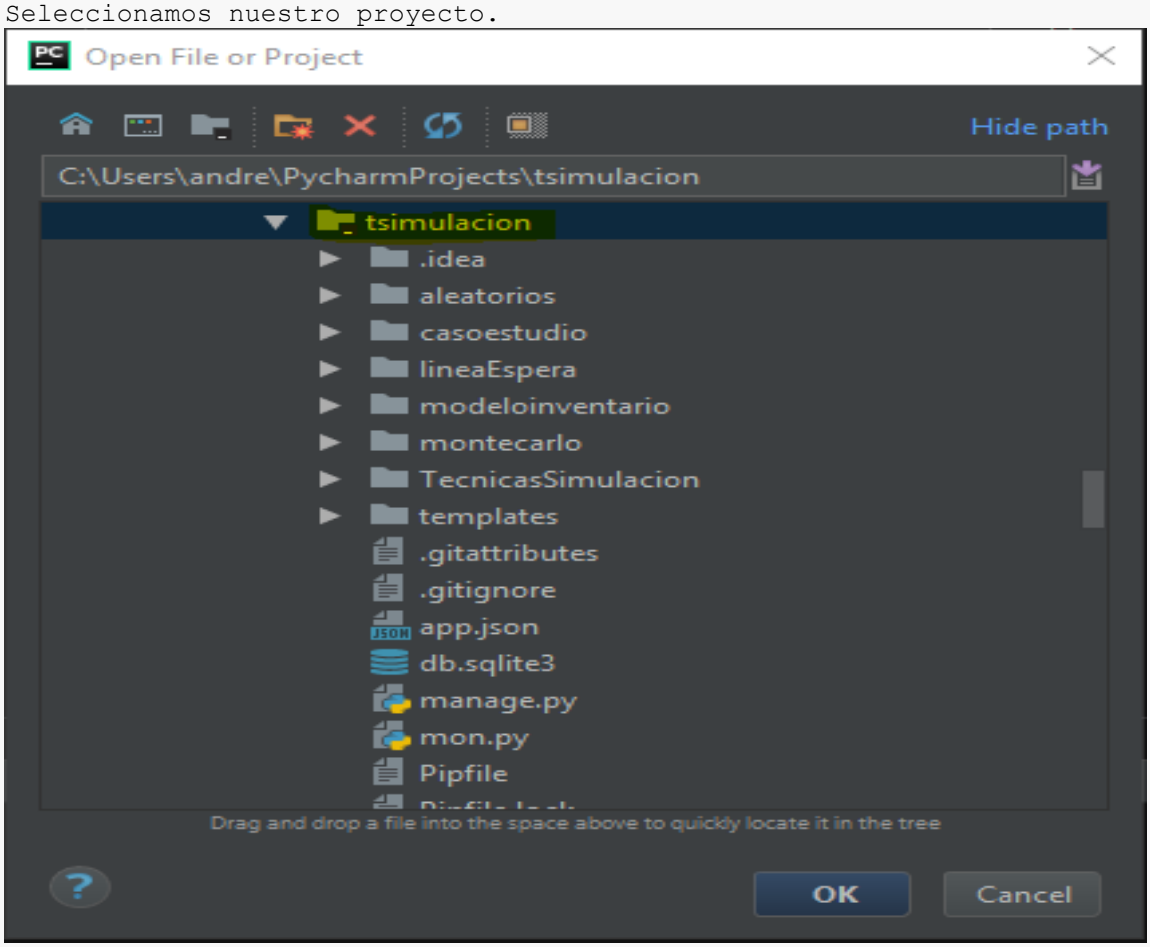
https://www.jetbrains.com/pycharm/documentation/

¿Cómo Iniciar este proyecto en su equipo con PyCharm?

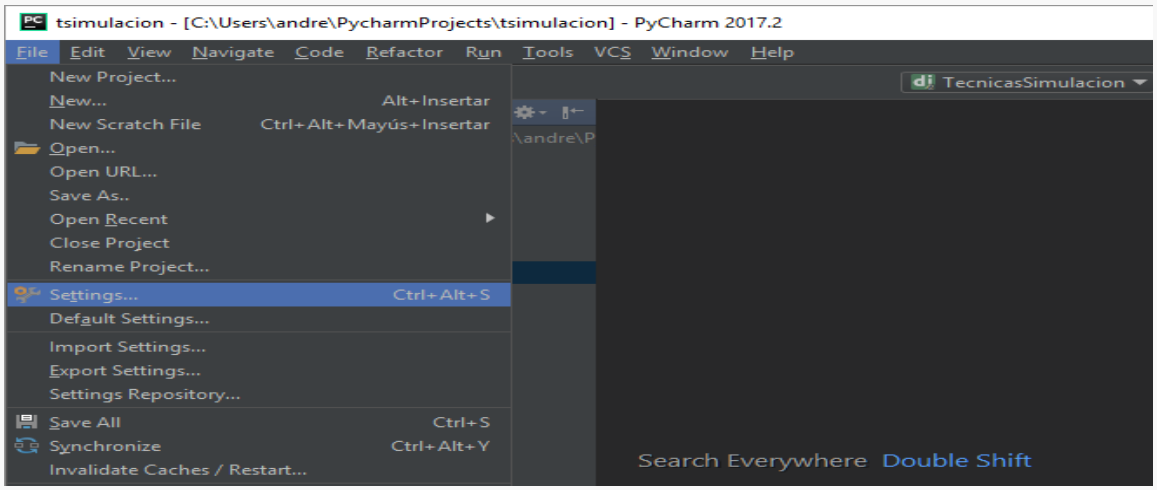
1. Iniciar PyCharm y presionar en Open.



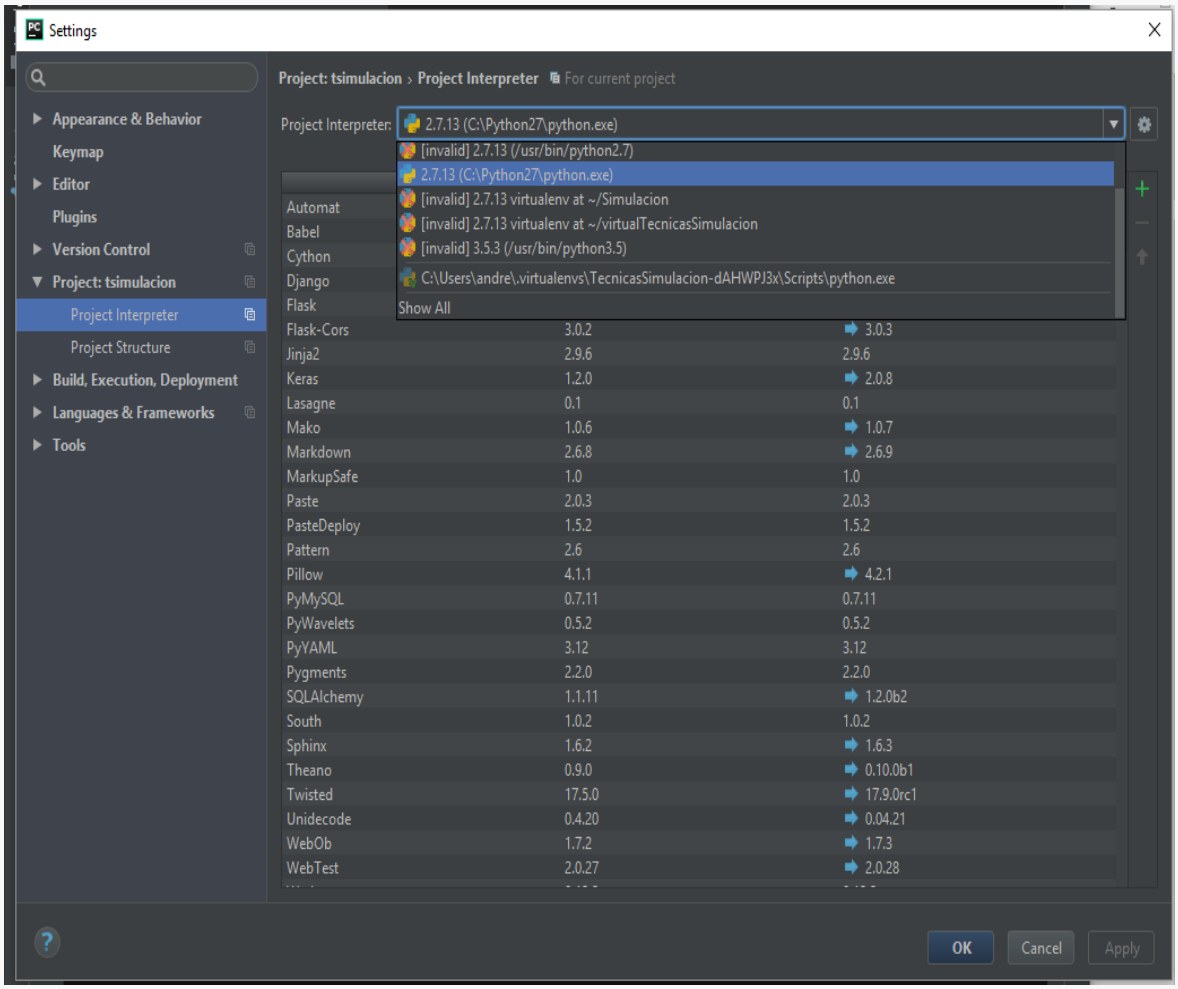
2. Seleccionamos nuestro proyecto.



3. Presionamos Ctrl + Alt + S o en File - Settings



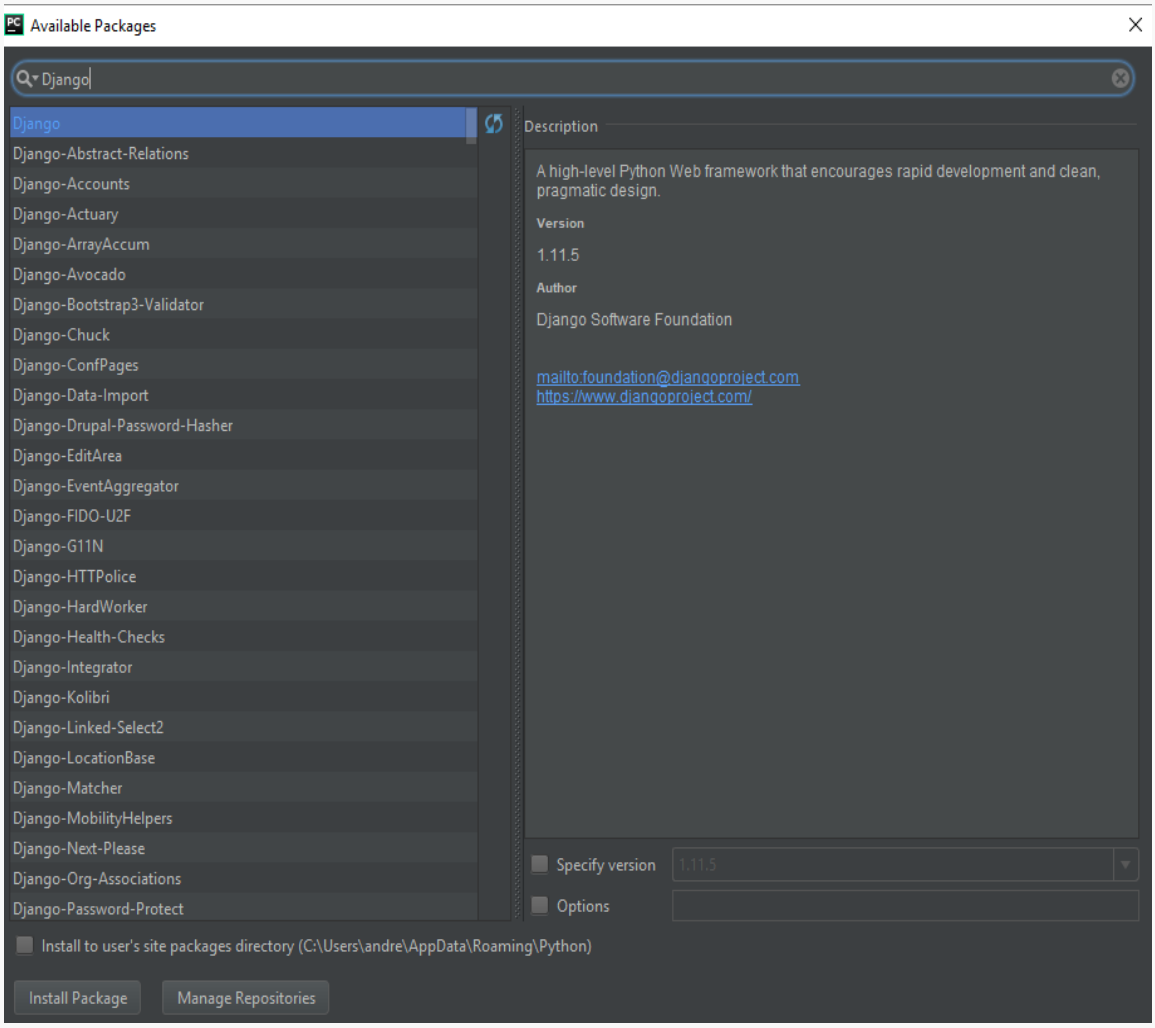
4. Nos ubicamos en Project > Project Interpreter y seleccionamos nuestro interprete de Python o creamos un Entorno Virtual.



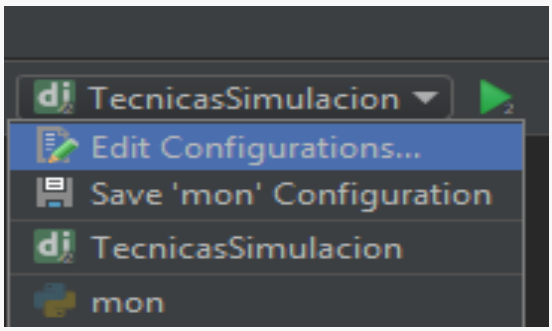
5. Presionamos en el Signo de + para agregar librerías.

6. Buscamos y Seleccionamos todas las Librerías necesarias en nuestro proyecto y presionamos en Install Package.

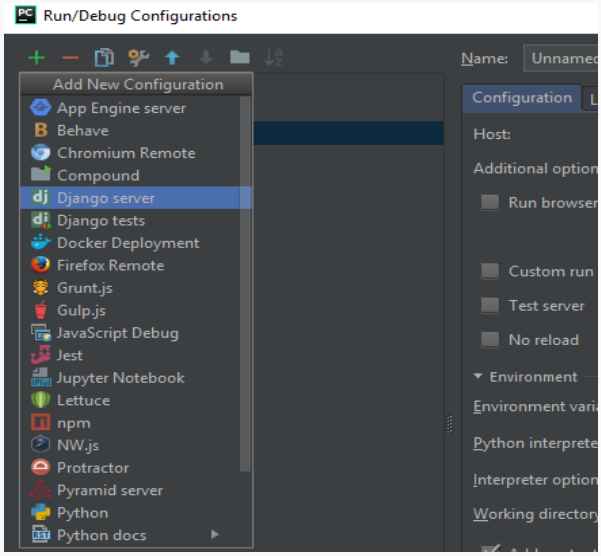
Django - Numpy – Scipy – Whitenoise - Pipenv



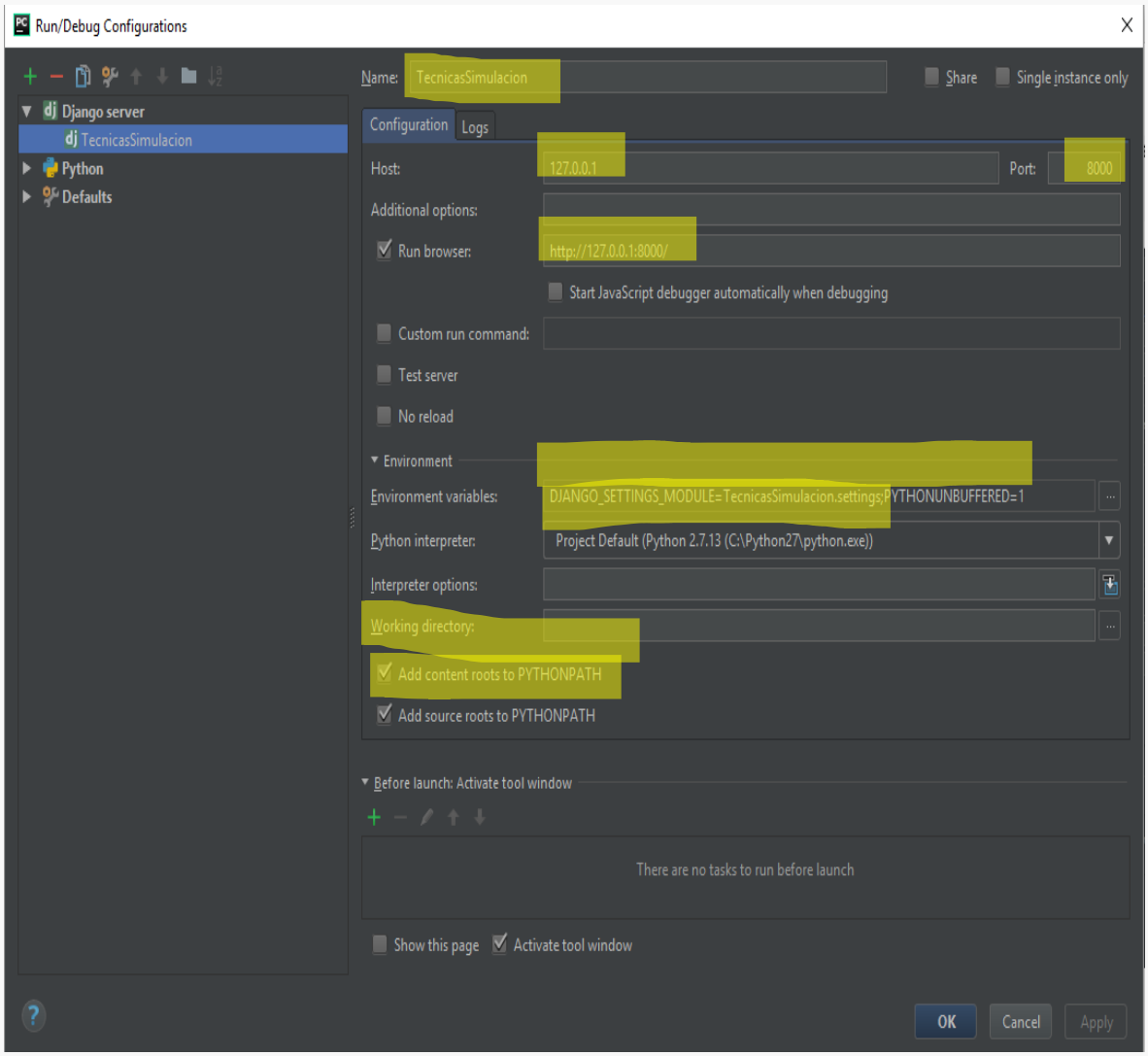
7. Cerramos el instalador de librerias y editamos las Configuraciones



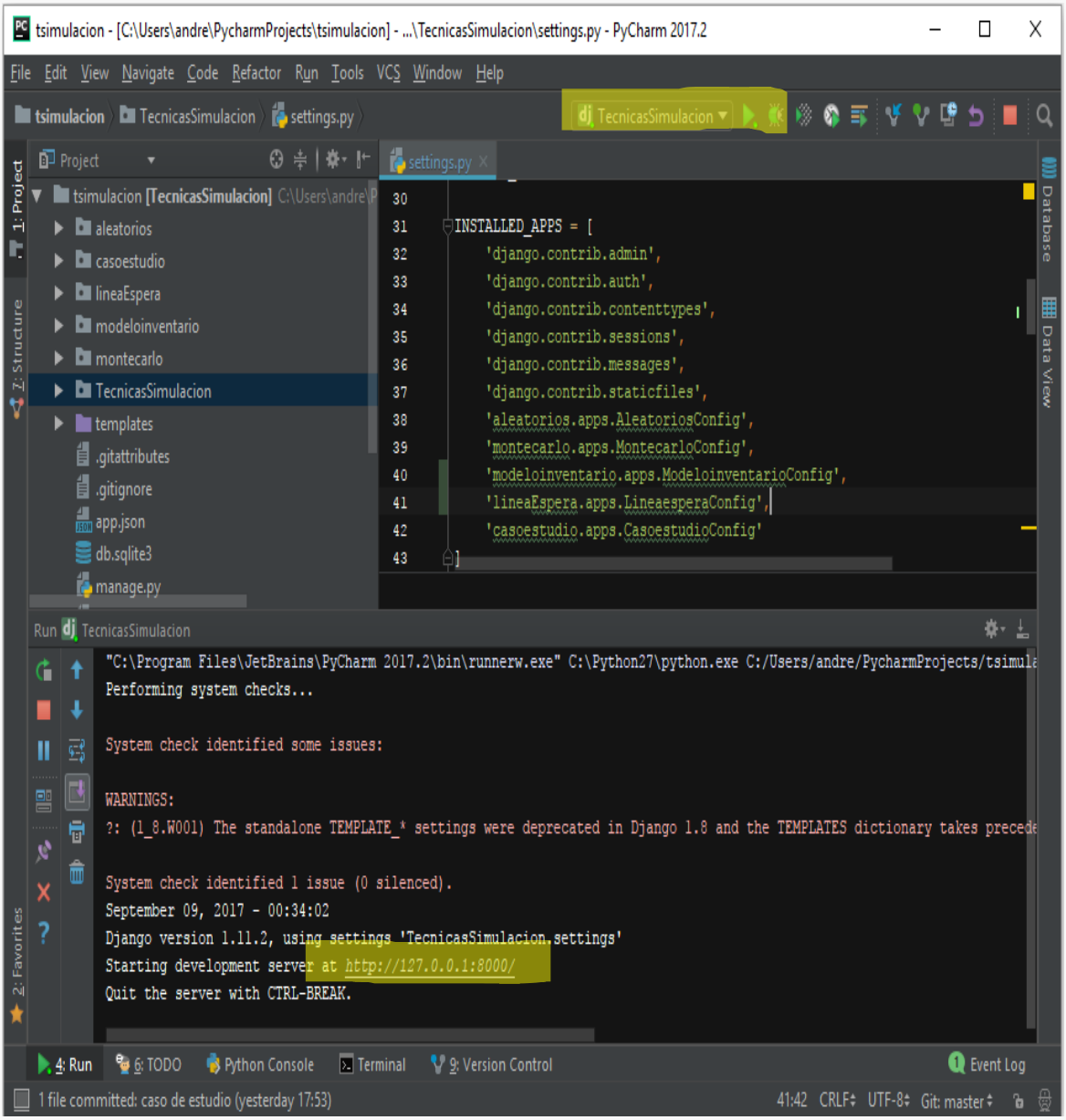
8. Si no hay una configuración de Django Server la agregamos y configuramos



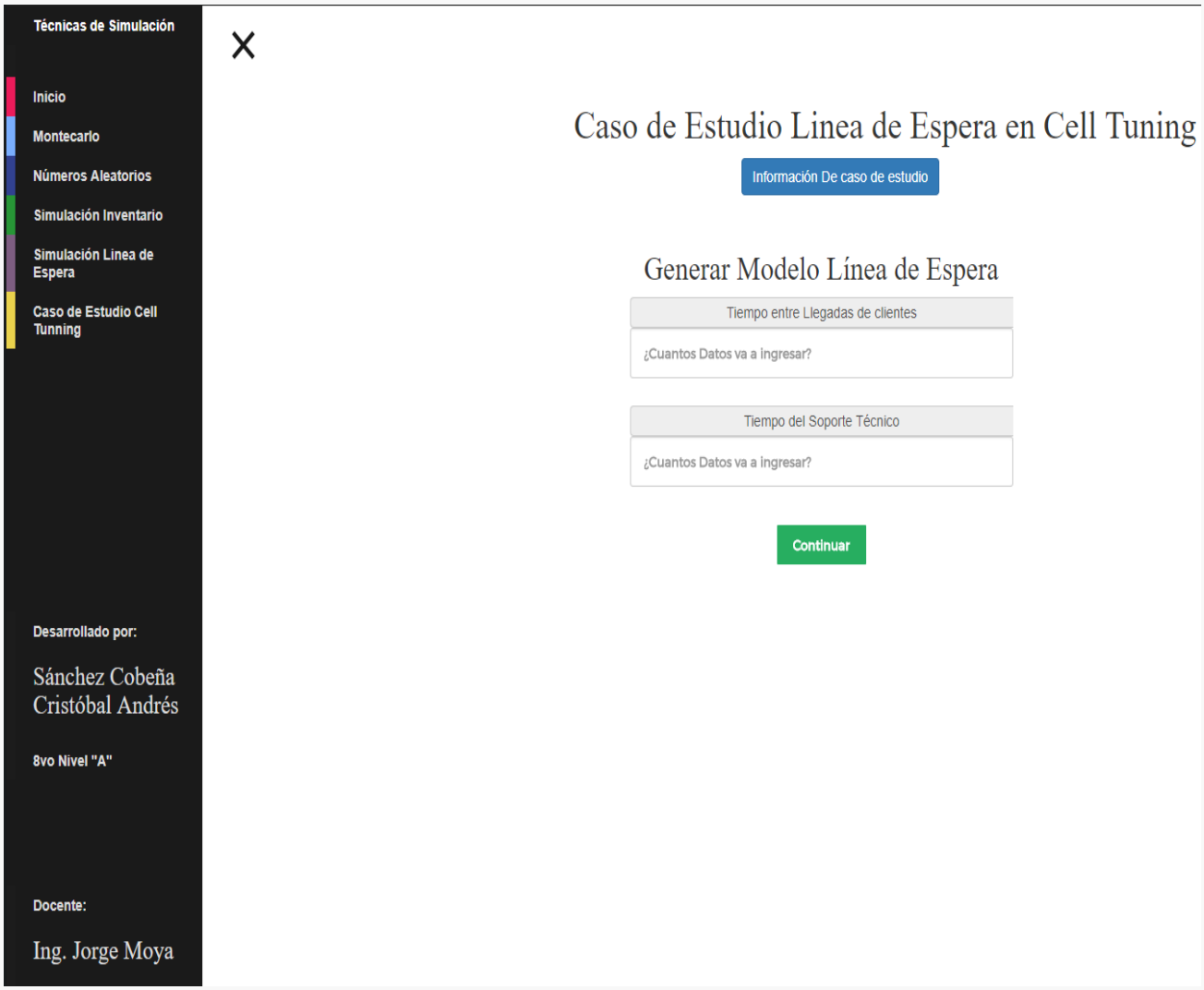
8.1 Configuración de Django Server



9. Guardamos y ejecutamos el Proyecto.



10. Visualización del Proyecto.



Primeros pasos con HEROKU

HEROKU CLI

Instrucciones Descargar e instalar:

https://devcenter.heroku.com/articles/heroku-cli#macos

MAC OS - HOMEBREW:

```
$ brew install heroku/brew/heroku
```

Compatible con 10.7 o versiones posteriores

MAC OS - INSTALADOR:

1. Descargar y ejecutar el instalador de MacOS .

Compatible con 10.7 o versiones posteriores

WINDOWS:

Descargar y ejecutar el instalador de Windows de 32 bits y de 64 bits .

1. Ejecute el archivo .exe que inicia el Asistente para la instalación.

2. Siga todos los pasos sugeridos por el asistente. Preste especial atención a las opciones de instalación correspondientes.

Debian / Ubuntu

Esta versión no se actualiza automática. Vas a tener que actualizar manualmente Heroku CLI con apt-get. Utilice la instalación independiente para una versión autoupdating de la CLI.

Ejecute el siguiente para agregar nuestro repositorio apt e instalar la CLI:

```
$ wget -qO- https://cli-assets.heroku.com/install-ubuntu.sh | sh
```

¿Como subirlo a un Servidor (Heroku)?

Documentación Oficial:

https://devcenter.heroku.com/articles/getting-started-with-python#introduction

Requisitos:

1. Crear una Cuenta en HEROKU

https://signup.heroku.com/dc

2. Instalar Python

3. Instalar PipEnv

```
pip install pipenv
```

4. Tener instalado HEROKU CLI

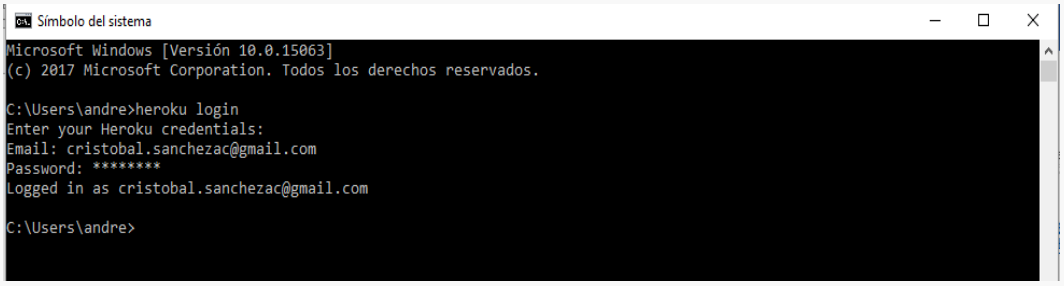
Preparar:

Una vez instalado, puede utilizar heroku con su shell de comandos.

1. Iniciamos nuestra consola e iniciamos sesión con el comando:

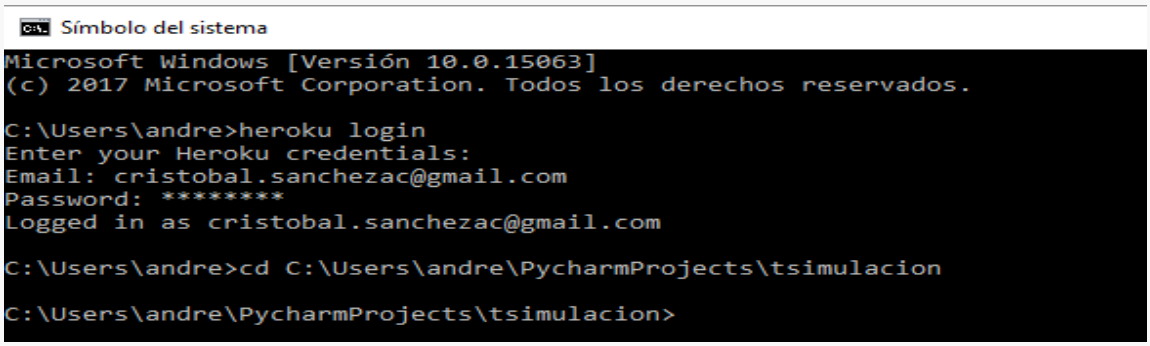
```
Heroku login
```

y ponemos nuestras credenciales creadas en heroku.



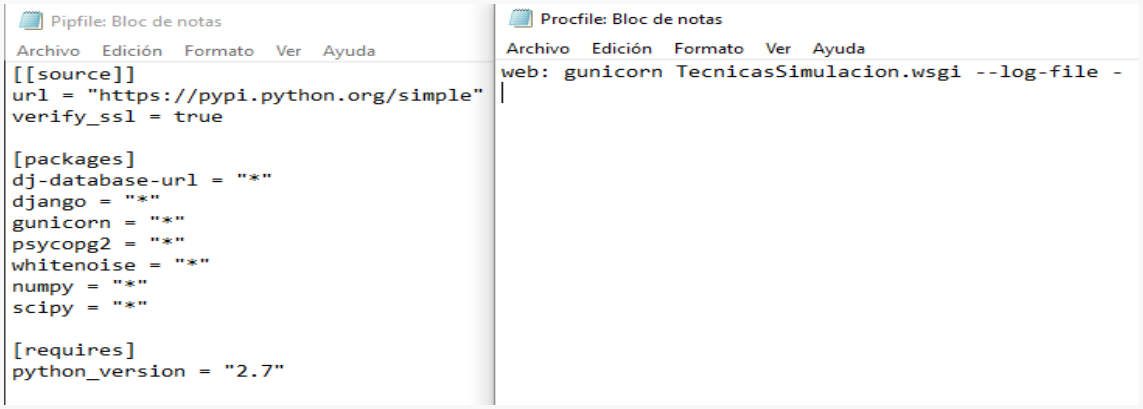
La autenticación es necesaria para permitir que los comandos heroku y git funcionen.

2. Nos ubicamos en la carpeta donde está el proyecto.



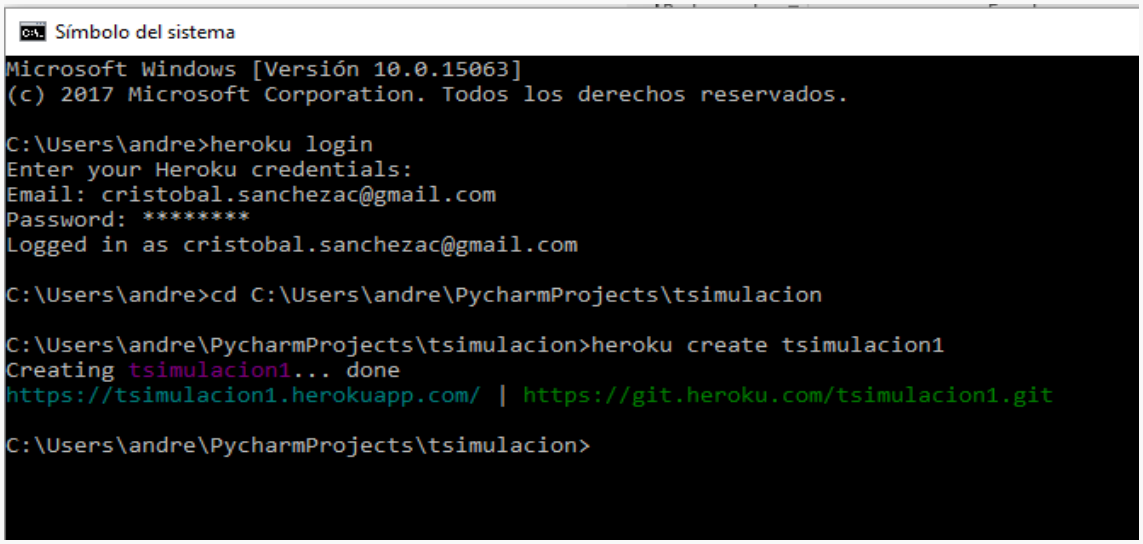
3. verificamos nuestros archivos Pipfile y Procfile

En el archivo PipFile van los nombres de las librerías utilizadas ya que de aquí toma la información para instalar las librerías necesarias para nuestro proyecto.



4. Creamos la aplicación, y también se nos crea un git.

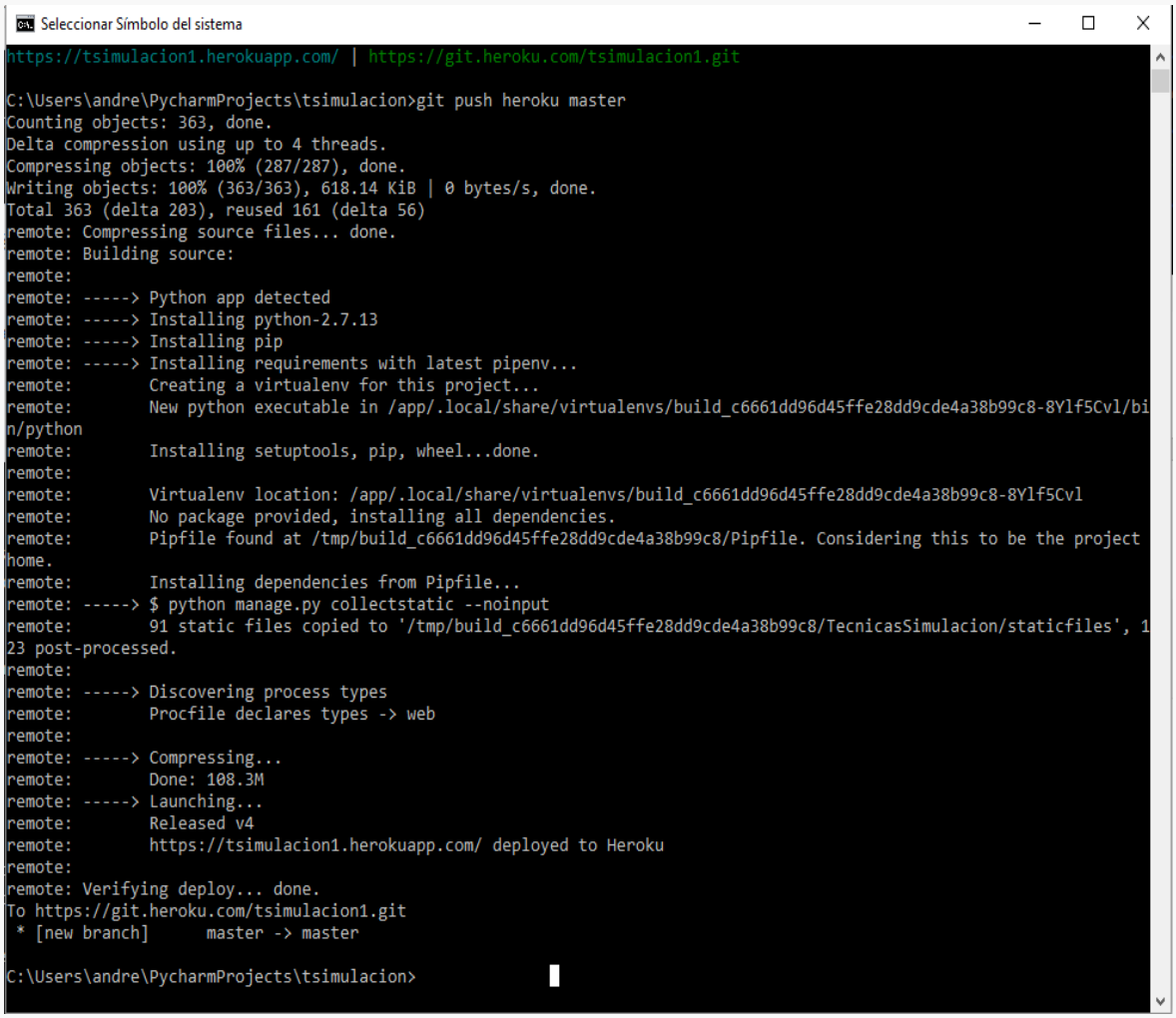
```
Heroku create nombredeAplicacion
```



5. Subimos los archivos de nuestro proyecto con git push.

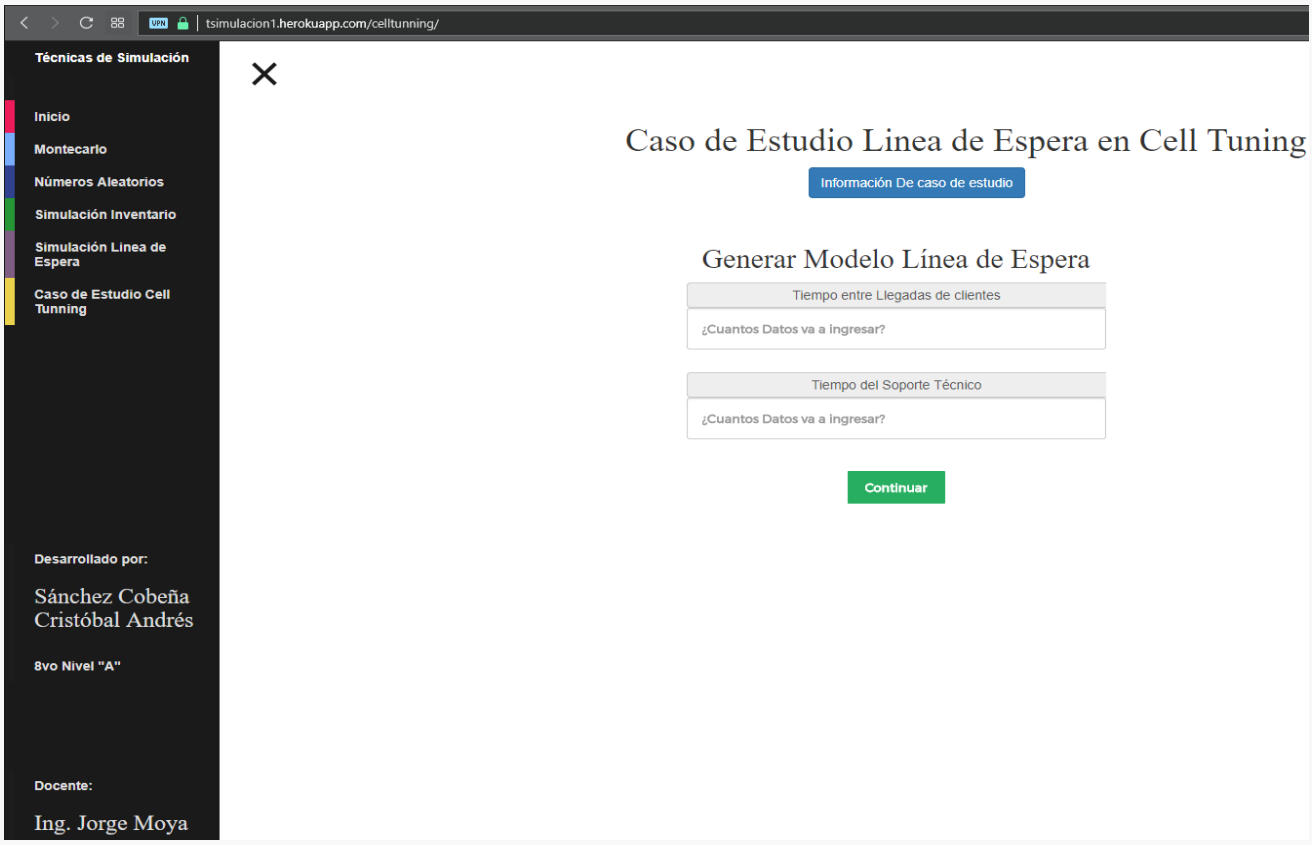
```
git push heroku master
```

Se instalan las librerías y se suben todos los archivos al servidor de heroku.



6. Verificamos que la página funcione correctamente y la abrimos con:

```
Heroku open
```



7. Si queremos ver los logs utilizamos el comando:

```
heroku logs --app tsimulacion1
```

```
C:\Users\andre\PycharmProjects\tsimulacion>heroku logs --app tsimulacion1
2017-09-09T06:22:28.000000+00:00 app[api]: Build succeeded
2017-09-09T06:23:37.030519+00:00 heroku[web.1]: Starting process with command `gunicorn TecnicasSimulacion.wsgi --log-fi
le -`
2017-09-09T06:23:39.556898+00:00 app[web.1]: [2017-09-09 06:23:39 +0000] [4] [INFO] Starting gunicorn 19.7.1
2017-09-09T06:23:39.558552+00:00 app[web.1]: [2017-09-09 06:23:39 +0000] [4] [INFO] Using worker: sync
2017-09-09T06:23:39.558338+00:00 app[web.1]: [2017-09-09 06:23:39 +0000] [4] [INFO] Listening at: http://0.0.0.0:23066 (
4)
2017-09-09T06:23:39.566910+00:00 app[web.1]: [2017-09-09 06:23:39 +0000] [9] [INFO] Booting worker with pid: 9
2017-09-09T06:23:39.583876+00:00 app[web.1]: [2017-09-09 06:23:39 +0000] [10] [INFO] Booting worker with pid: 10
2017-09-09T06:23:40.613741+00:00 heroku[web.1]: State changed from starting to up
2017-09-09T06:25:25.535812+00:00 heroku[router]: at=info method=GET path="/" host=tsimulacion1.herokuapp.com request_id=
ad438487-38f2-4f18-acfa-984a04d64cc7 fwd="190.214.154.197" dyno=web.1 connect=1ms service=321ms status=200 bytes=12282 p
rotocol=https
2017-09-09T06:25:25.703510+00:00 heroku[router]: at=info method=GET path="/static/css/estilo.css" host=tsimulacion1.hero
kuapp.com request_id=179ba008-a293-4618-b455-b0dc11a353ae fwd="190.214.154.197" dyno=web.1 connect=2ms service=3ms statu
s=200 bytes=4038 protocol=https
2017-09-09T06:25:25.707092+00:00 heroku[router]: at=info method=GET path="/static/css/bootstrap.css" host=tsimulacion1.h
erokuapp.com request_id=530b6162-8414-4cf1-9f39-0bc724cecae8 fwd="190.214.154.197" dyno=web.1 connect=1ms service=14ms s
tatus=200 bytes=154853 protocol=https
2017-09-09T06:25:26.046943+00:00 heroku[router]: at=info method=GET path="/static/js/bootstrap.js" host=tsimulacion1.her
okuapp.com request_id=250aad67-94da-4999-b2d8-3d4a4c147bab fwd="190.214.154.197" dyno=web.1 connect=1ms service=8ms stat
us=200 bytes=76836 protocol=https
2017-09-09T06:25:25.898233+00:00 heroku[router]: at=info method=GET path="/static/css/style.css" host=tsimulacion1.herok
uapp.com request_id=ff7733dc-aba3-4676-a921-5c240ef7023d fwd="190.214.154.197" dyno=web.1 connect=1ms service=3ms status
=200 bytes=3507 protocol=https
2017-09-09T06:25:26.077546+00:00 heroku[router]: at=info method=GET path="/static/js/valid.js" host=tsimulacion1.herokua
pp.com request_id=f8fe899e-7c77-4ad0-9d5a-fc5c4fec611f fwd="190.214.154.197" dyno=web.1 connect=1ms service=4ms status=2
00 bytes=21724 protocol=https
2017-09-09T06:25:26.065092+00:00 heroku[router]: at=info method=GET path="/static/js/bootstrap.min.js" host=tsimulacion1
.herokuapp.com request_id=e7824ba8-92c2-4148-96db-42e4d61857d7 fwd="190.214.154.197" dyno=web.1 connect=1ms service=5ms
status=200 bytes=37364 protocol=https
2017-09-09T06:25:26.044676+00:00 heroku[router]: at=info method=GET path="/static/js/funciones.js" host=tsimulacion1.her
```

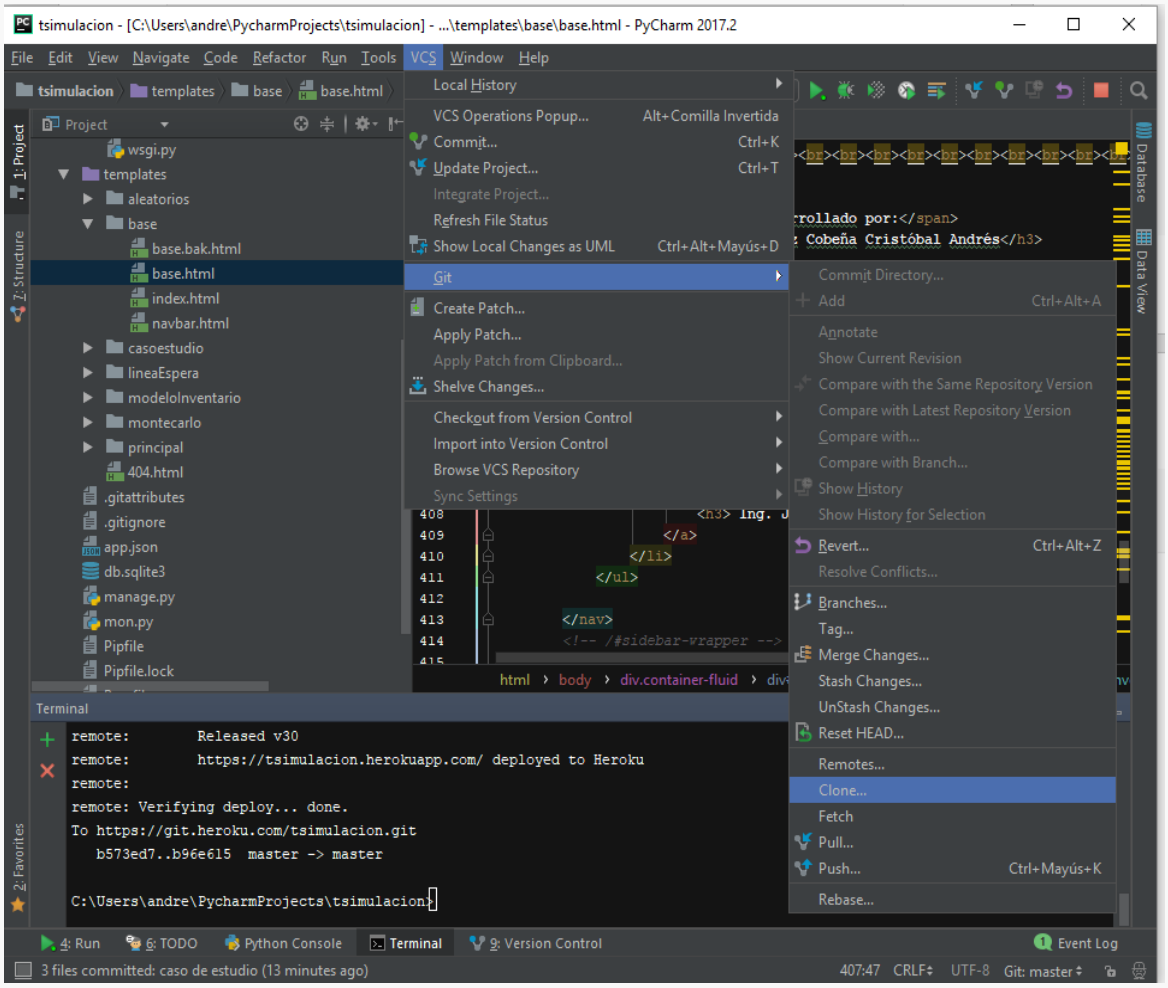
¿Como modificar nuestro proyecto subido en Heroku?

1. Iniciamos pycharm

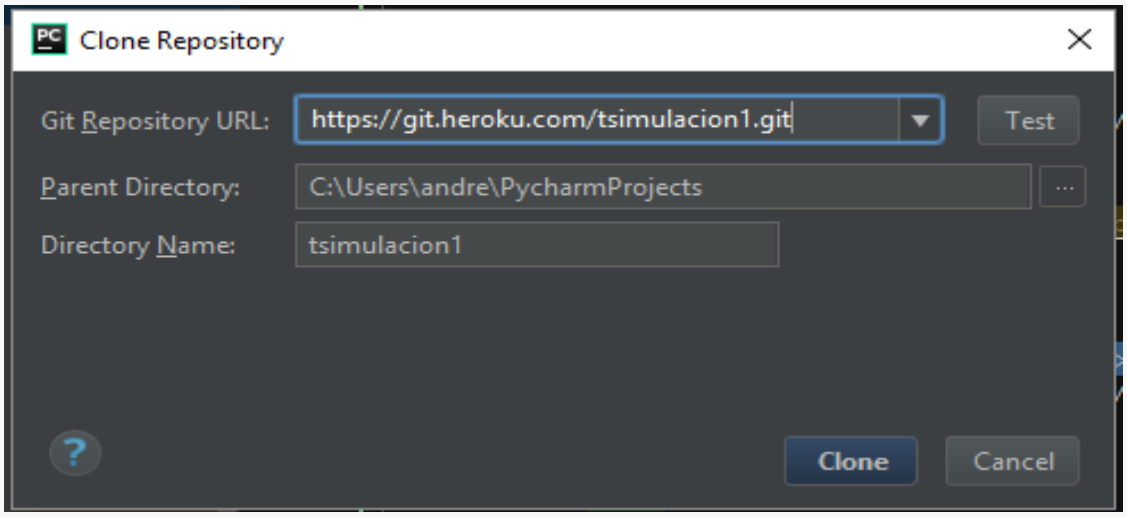
2. Nos dirigimos a VCS y hacemos click en enable repository

3. Nuevamente vamos a VCS – git – y clone, clonaremos nuestro proyecto, en este caso nuestro repositorio que nos creó fue:

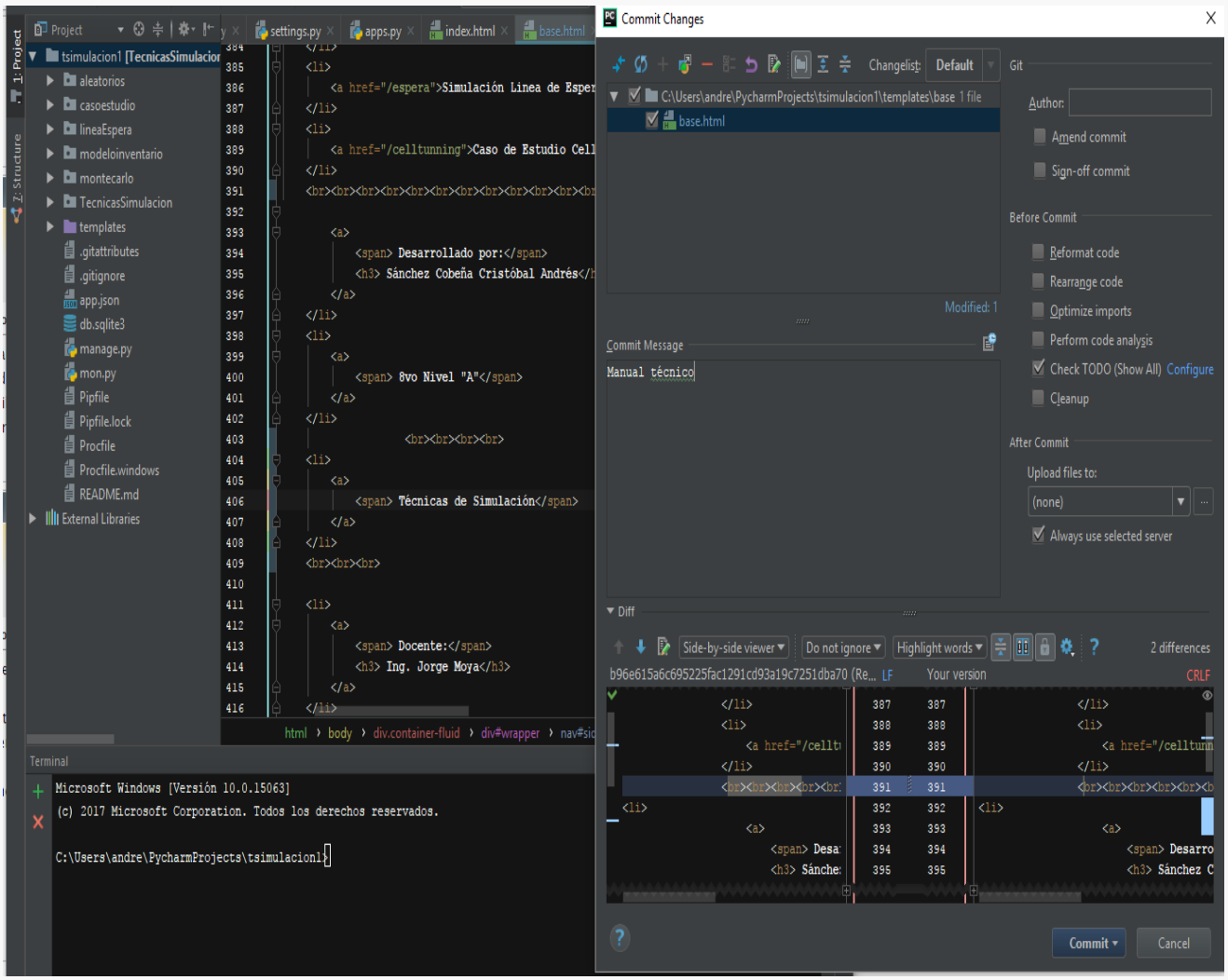
    https://git.heroku.com/tsimulacion1.git



4. Clonamos nuestro proyecto pegando el enlace aquí.

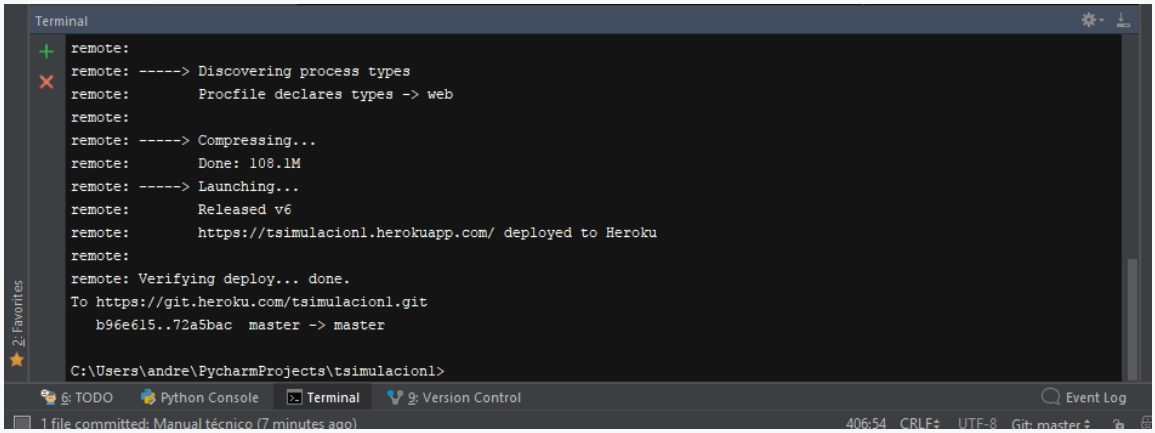


5. Una vez copiado nuestros archivos de git en nuestros archivos locales podemos editarlos, para guardar los cambios hacemos un commit, y escribimos algún comentario.



6. hacemos un git push para actualizar los archivos al git y se actualice nuestra página.

```
git push
```



7. Verificamos los cambios en nuestra página.