

```
ssh usryates@dev.yatestogo.com  
python manage.py collectstatic --settings=config.settings.production
```

```
ssh yates@mt.rady.com.co  
activar entorno virtual: source env_yates/bin/activate
```

```
python manage.py runserver --settings=config.settings.dev_isabel  
python manage.py shell_plus --settings=config.settings.dev_isabel  
python manage.py runserver_plus --settings=config.settings.dev_isabel  
python manage.py collectstatic --settings=config.settings.dev_test
```

PARA PRUEBAS

```
--settings=config.settings.dev_test
```

PARA PRODUCCION

```
--settings=config.settings.production
```

```
http://localhost:8000/rental/confirmation-reservation/f4iJNaHp/  
mensaje de reserva expirada en esa url de confirmacion
```

```
python manage.py makemigrations --settings=config.settings.production
```

```
python manage.py loaddata data_ --settings=config.settings.production
```

```
python manage.py migrate --settings=config.settings.production
```

```
python manage.py collectstatic --settings=config.settings.production
```

Acceso al servidor de pruebas

Tipo de acceso: SSH/SFTP

Servidor: mt.rady.com.co

Usuario: yates

forma de acceder:

ssh yates@mt.rady.com.co

activar entorno virtual: `source env_yates/bin/activate`

Servidor de BD: **bd.local**

Usuario: **kevinusr**

Contraseña: **bhjay72hyufa783**

DB: **yatesdev**

URL de administración: <http://mt.rady.com.co/mantenimiento/admindb/>

Acceso al servidor de Producción:

Nombre servidor: dev.yatestogo.com

URL aplicación: <https://dev.yatestogo.com>

Usuario / Contraseña de SSH: usryates / TorcazaCambiaTurno2018

Rutas en disco para la aplicación:

Logs en tiempo real: /var/www/app/log_db/

ruta para el montaje de la aplicación: /var/www/app

ruta para configuracion: /var/log/httpd

acceso:

ssh usryates@dev.yatestogo.com

ingreso por ssh

Datos de la base de datos:

Usuario: yatesusr

Contraseña: YwSX5CDNZRag7GhT

DB: yatesdb

URL de administración: <https://dev.yatestogo.com/mantenimiento/admindb/>

PASOS PARA LEVANTAR EL SERVICIO:

```
nohup python manage.py runserver 45.58.40.92:8089  
--settings=config.settings.dev_test
```

COMANDOS

```
python manage.py runserver --settings=config.settings.dev_test  
python manage.py makemigrations --settings=config.settings.dev_test  
python manage.py migrate --settings=config.settings.dev_test  
python manage.py collectstatic --settings=config.settings.dev_test
```

PARA PRUEBAS

```
--settings=config.settings.dev_test
```

PARA PRODUCCIÓN

```
--settings=config.settings.production
```

```
python manage.py collectstatic --settings=config.settings.production
```

Instalar tarea automática:

```
python manage.py installtasks --settings=config.settings.dev_test
```

Generar backup:

```
python manage.py dumpdata app_name --format yaml > file.yaml --settings=config.settings.production
```

Leer backup:

```
python manage.py loaddata file.yaml --settings=config.settings.production
```

```
python manage.py loaddata data_scripts/1_pages_information.yaml --settings=config.settings.production
```

Descargar backup

```
scp -r usryates@dev.yatestogo.com:/var/www/app/yatestogo/yates2go/db.yaml
```

Ayuda:

```
$ ./manage.py -h
```

Bajar una carpeta del servidor:

```
:~$ scp -r usryates@dev.yatestogo.com:/var/www/app/yatestogo/yates2go .
```

```
:~$ scp -r usryates@dev.yatestogo.com:/var/www/app/yatestogo/yates2go/
```

Subir un archivo al servidor:

```
:~$ scp /home/rady/PROJECT-YATES/yatestogo/yates2go/media/default.jpg  
usryates@dev.yatestogo.com:/var/www/app/yatestogo/yates2go/media
```

```
scp ruta/img.jpg usryates@dev.yatestogo.com:ruta_destino
```

Subir carpeta media al servidor de pruebas:

```
scp -pr /home/rady/PROJECT-YATES/yatestogo/yates2go/media/  
yates@mt.rady.com.co:/home/yates/project/yatestogo/yates2go/
```

```
scp -pr /home/rady/PROJECT-YATES/media/  
usryates@dev.yatestogo.com:/var/www/app/yatestogo/yates2go/
```

cuando trabajo en ramas remotas:

1. git fetch
2. luego pasarme a la rama remota

```
python manage.py shell < data_scripts/populate_django_cities.py  
--settings=config.settings.production
```

Ambiente servidor de Pruebas:

- para servidor test:
- \$ mkvirtualenv -p /usr/local/bin/python env_yates
- source yates/bin/activate

Crear ambiente virtual:

Instalar ambiente:

- \$ pip install virtualenvwrapper
- \$ sudo pip install virtualenvwrapper
-

Crear ambiente:

- \$ mkvirtualenv -p /usr/bin/python3 env_yates
- en windows: mkvirtualenv ambiente
- \$ workon name_env
- \$ deactivate
- VS
- \$ pip install --upgrade virtualenv
- \$ virtualenv env_yates
- \$ source env_yates/bin/activate
- nota: env_yates es el nombre del ambiente virtual

Eliminar ambiente virtual:

- \$ rmvirtualenv ENVNAME -- con virtualenv wrapper
- rm -r fichero_o_carpeta_a_eliminar -- para virtualenv basta con borrar la carpeta del directorio que contiene el ambiente virtual
- rm -rf <directorio> (forzada)

Listar ambientes virtuales existentes:

- \$ lsvirtualenv -b

Copiar un entorno existente:

- \$ cpvirtualenv ENVNAME [TARGETENVNAME]

Llamar a este comando para mostrar el contenido del directorio de paquetes de sitio del virtualenv activo:

- \$ lssitepackages

Ver los paquetes instalados:

- \$ pip freeze

Cambiar entre ambientes con workon:

- (env2)\$ workon env1
- (env1)\$ echo \$VIRTUAL_ENV
- /Users/dhellmann/Envs/env1
- (env1)\$

Comandos:

- instalar software por consola de ubuntu:
 - \$ sudo dpkg -i slack-desktop-3.0.5-amd64.deb
- Instalar controlador de Postgres:
 - \$ pip install django psycopg2

Exportar importar data:

Crear carpeta 'data_scripts' en la misma ubicación de manage.py

```
$ ./manage.py dumpdata
```

```
$ ./manage.py dumpdata -h
```

```
$ ./manage.py loaddata
```

```
./manage.py dumpdata >>my_backup.json
```

```
./manage.py loaddata my_backup.json
```

<https://coderwall.com/p/mvsoyg/django-dumpdata-and-loaddata>

Consola interactiva de python

```
./manage.py shell
```

Notas para operaciones de data en consola:

<http://django-book.blogspot.com.co/2012/11/relaciones-many-to-many.html>

```
from products.models import Service, Product
>>> for e in Service.objects.all():
...     print(e.name)
...
```

```
>>> for yate in Product.objects.all():
...     if yate.max_capacity > 5:
...         print(yate.name)
```

```
>>> for yate in Product.objects.all():
...     list_yates.append(yate.name)
...
>>> list_yates
['Marlin', 'Sea Devil', 'Dog House', 'Matin Pescador', 'Formula', 'Fivres']
```

Para borrar data:

```
$ python manage.py flush
```

Para cargar data desde una sub carpeta:

```
$ python manage.py shell < data_scripts/initial.py
```

Proceso subir a servidor producción

- ssh usryates@dev.yatestogo.com
- cd /var/www/app/yatestogo/yates2go/ (no hay entornos virtuales)
- Para el servidor:
 - apachestop
- Trabajar siempre en la rama master
 - git pull origin master
- correr migraciones
- migrar
- correr collectstatic
- levantar servidor
 - apachestart

Nota: Pasos para cambiar usuario de git

- Donde esta el archivo README.md dejar la carpeta git visible
 - mv .git git
- Entrar a la carpeta git y buscar el archivo de configuración
 - nano config
- cambiar los parámetros de conexión, usuario y remoto
- volver a dejar la carpeta oculta
 - mv git .git