

# Como instalar TurboGears 2 en tu plan de alojamiento

En esta guía te explicamos como usar el framework Turbogears 2 en guebs.

## 1 - Crear la aplicación desde el panel de Hosting

Para instalar [Turbogears](#) lo primero que debes hacer es definir una aplicación desde el panel de hosting del plan de alojamiento.

El nombre que pongas como nombre de aplicación será el nombre de tu aplicación Turbogears. Supondremos que has elegido **miapp** como nombre de aplicación.



**Crear nueva aplicación python**

Nombre de la aplicación:   
💡 Se usará como nombre de directorio de la aplicación, nada de caracteres raros.

Dominio:

Stack:   
💡 Se permite un único stack por cuenta de hosting, el asignado a esta es **python-3.4**

La elección de nombre **miapp** implica que el directorio de tu aplicación Turbogears será **python/miapp**

## 2 - Instalar Turbogears

Una vez hayamos creado la aplicación desde el panel de hosting, sólo debemos entrar en sesión ssh y realizar las siguientes acciones:

<br />

```
cd python<br />
rmdir miapp<br />
pip install tg.devtools &#8211;user<br />
~/local/bin/gearbox quickstart miapp<br />
```

Esto creará un esqueleto mínimo para la aplicación, que estará ubicado en **python/miapp**.

Debemos crear el directorio **public** dentro de la misma, que es donde almacenaremos los archivos estáticos de la aplicación:

```
<br />
cd miapp<br />
mkdir public<br />
```

Además, copiaremos el archivo development.ini a **production.ini**:

```
<br />
cp -a development.ini production.ini<br />
```

### 3 - Instalar dependencias

Instalaremos las dependencias de TurboGears:

```
<br />
pip install -e . &#8211;user<br />
```

Si usas python 3.4, existe un problema entre genshi y las últimas versiones de TurboGears que produce una excepción en la ejecución de la aplicación. Para evitarlo debemos añadir la siguiente directiva en nuestros archivos .ini, en la sección [app:main]

```
templating.genshi.name_constant_patch = true
```

El error que verías en caso de no añadirlo es éste:

```
Exception: Unhandled node type <class '_ast.NameConstant'>
```

Fuente:

<http://turbogears.readthedocs.io/en/latest/cookbook/upgrading.html#genshi-work>

-around-available-for-python3-4

## 4 - Configurar el archivo WSGI

Suponiendo que el usuario de tu hosting es **usuario\_cpanel** y dado que el nombre de la aplicación que hemos definido es **miapp**, creamos el archivo **passenger\_wsgi.py** con el siguiente contenido:

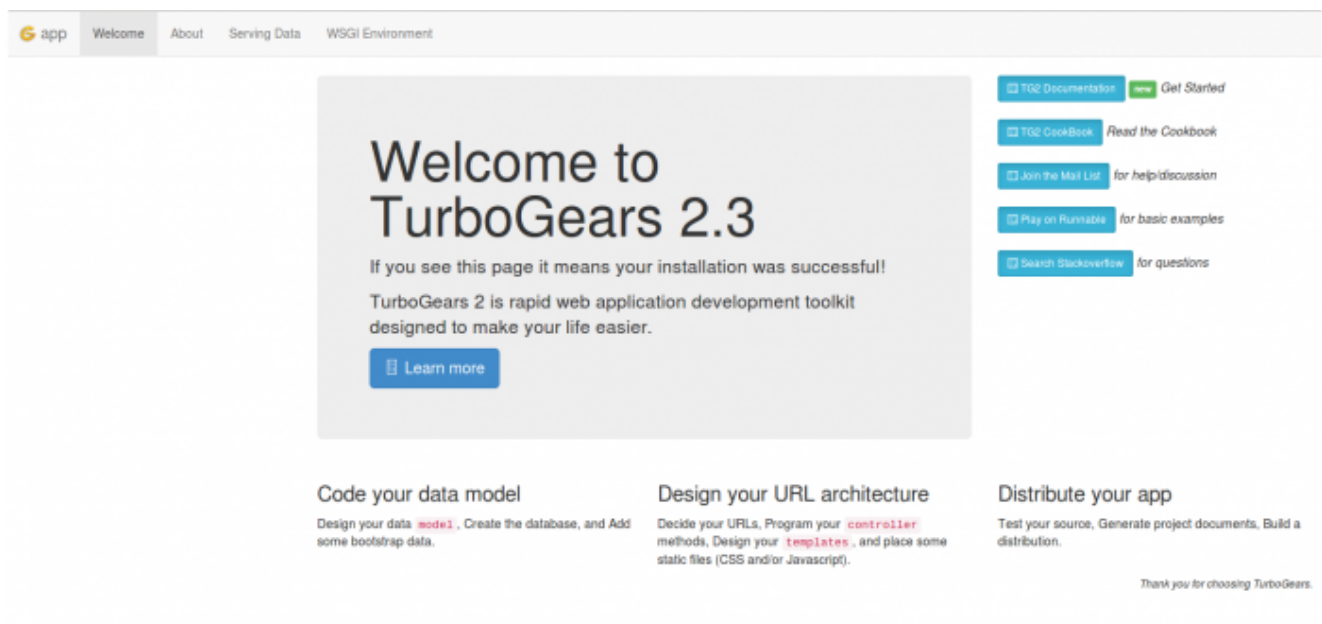
```
APP_CONFIG =
"/home/usuario_cpanel/python/miapp/production.ini"

#Setup logging
import logging.config
logging.config.fileConfig(APP_CONFIG)

#Load the application
from paste.deploy import loadapp
application = loadapp('config:%s' % APP_CONFIG)
```

## 5 - !Listo!

Y con esto veremos ya la página por defecto de TurboGears funcionando:



app Welcome About Serving Data WSGI Environment

# Welcome to TurboGears 2.3

If you see this page it means your installation was successful!

TurboGears 2 is rapid web application development toolkit designed to make your life easier.

[Learn more](#)

[T22 Documentation](#) [Get Started](#)

[T22 Cookbook](#) [Read the Cookbook](#)

[Join the Mail List](#) [for help/discussion](#)

[Play on Runable](#) [for basic examples](#)

[Search Stackoverflow](#) [for questions](#)

### Code your data model

Design your data **model**, Create the database, and Add some bootstrap data.

### Design your URL architecture

Decide your URLs, Program your **controller** methods, Design your **templates**, and place some static files (CSS and/or Javascript).

### Distribute your app

Test your source, Generate project documents, Build a distribution.

Thank you for choosing TurboGears.