Diplomatura en Django:

**Curso de Django Nivel Avanzado – En producción**

Módulo 2:

**Configuraciones finales**

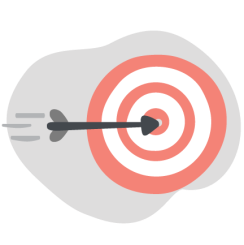
Unidad 3:

**SEO**

# 

### Presentación

En esta unidad haremos los ajustes finales de llevar un trabajo a producción.



### Objetivos

**Que los participantes logren…**

Comprender cuales son los aspectos necesarios para llevar el trabajo a producción.

# 

### Bloques temáticos

1. Ckeditor
2. django-simple-history
3. https en producción

# 1.- Editor de texto en TextField

**WYSIWYG** es el acrónimo de What You See Is What You Get (en español, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

Opciones: **CKEditor**

<http://ckeditor.com/download>

## PASO 1 - Agregar ckeditor al path

pip install django-ckeditor

## PASO 2 – AGREGAR CKEDITOR A INSTALLED\_APPS

|  |
| --- |
| settings.py |
| INSTALLED\_APPS = [      ------      'ckeditor',      'ckeditor\_uploader',      # MIS APPS  -------  ] |

## PASO 3 – Agregar jquery

|  |
| --- |
| settings.py |
| # CKEDITOR  CKEDITOR\_JQUERY\_URL = '//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js' |

## PASO 4 - Ejecuto collectstatic

Ahora ejecuto:

|  |
| --- |
| python manage.py collectstatic |

## PASO 5 – urls.py

|  |
| --- |
| urls.py |
| urlpatterns = [  -------      include('ckeditor\_uploader.urls')),      ------  ] |

## PASO 6 - OPCIONAL

Configuración de ckeditor en settings.py:

|  |
| --- |
| settings.py |
| CKEDITOR\_UPLOAD\_PATH = "uploads/"  CKEDITOR\_CONFIGS = {      'default': {          'toolbar': (              ['div', 'Source', '-', 'Save', 'NewPage', 'Preview', '-', 'Templates'],              ['Cut', 'Copy', 'Paste', 'PasteText', 'PasteFromWord', '-', 'Print', 'SpellChecker', 'Scayt'],              ['Undo', 'Redo', '-', 'Find', 'Replace', '-', 'SelectAll', 'RemoveFormat'],              ['Form', 'Checkbox', 'Radio', 'TextField', 'Textarea', 'Select', 'Button', 'ImageButton', 'HiddenField'],              ['Bold', 'Italic', 'Underline', 'Strike', '-', 'Subscript', 'Superscript'],              ['NumberedList', 'BulletedList', '-', 'Outdent', 'Indent', 'Blockquote'],              ['JustifyLeft', 'JustifyCenter', 'JustifyRight', 'JustifyBlock'],              ['Link', 'Unlink', 'Anchor'],              ['Image', 'Update', 'Flash', 'Table', 'HorizontalRule', 'Smiley', 'SpecialChar', 'PageBreak'],              ['Styles', 'Format', 'Font', 'FontSize'],              ['TextColor', 'BGColor'],              ['Maximize', 'ShowBlocks', '-', 'About', 'pbckcode'],          ),      }  } |

## PASO 7 – models.py

Configuración de campo de texto en models.py:

|  |
| --- |
| models.py |
| from ckeditor\_uploader.fields import RichTextUploadingField  ------------  class PreguntasFrecuentes(models.Model):      -------  contenido = RichTextUploadingField('contenido') |

## PASO 8 – models.py

Configuración de vista:

|  |
| --- |
| preguntas\_frecuentes.html |
| <div class="container">      <div class="row">          <div class="col-12" >              {% for pregunta in preguntas %}                  <h1>{{pregunta.pregunta}}</h1>                  <p>{{pregunta.descripcion}}</p>                  {{pregunta.contenido|safe}}              {%endfor%}          </div>      </div>  </div> |

# 2 django-simple-history

## PASO 1 – Instalación

|  |
| --- |
| pip install django-simple-history  sudo pip3 install django-simple-history |

## PASO 2 – Configuración

Configuramos settings.py

|  |
| --- |
| settings.py |
| INSTALLED\_APPS = [      -----------------      'simple\_history',      -----------------  ]  MIDDLEWARE = [     ---------  'simple\_history.middleware.HistoryRequestMiddleware',  ] |

## PASO 3 – Configuramose el models.py

|  |
| --- |
| models.py |
| from simple\_history.models import HistoricalRecords  class PreguntasFrecuentes(models.Model):      -------      history = HistoricalRecords() |

## PASO 4 – Migramos los datos

|  |
| --- |
| python manage.py makemigrations  python manage.py migrate |

# 3 https en produccion

## PASO 1 – Habilitar ssl

|  |
| --- |
| a2enmod ssl  systemctl restart apache2 |

## PASO 2 – Miro si el puerto 443 está abierto

**2.1 Instalo netstat**

|  |
| --- |
| sudo apt install net-tools |

**2.2 Veo si el puerto está habilitado**

nc -zv {IP} {PUERTO}

nc -zv {IP} 443

Si está habierto da: succeeded!

Si está cerrado da: Connection refused

**2.3. Si está cerrado puedo abrirlo así:**

|  |
| --- |
| sudo ufw allow 80  sudo ufw allow 443 |

ó

|  |
| --- |
| sudo ufw allow http  sudo ufw allow https |

## PASO 3 – Reinicio apache

|  |
| --- |
| sudo systemctl restart apache2 |

## PASO 4

Vamos a:

|  |
| --- |
| cd /etc/apache2/sites-available |

Ingresamos a 000-default.conf y cambiamos esta línea que está al inicio:

<VirtualHost \*:80>

Por:

000-default.conf

<VirtualHost dominio:80>

## PASO 5

Vamos a:

|  |
| --- |
| cd /etc/apache2/sites-available |

E ingresamos a: default-ssl.con

Copiamos todo lo que tiene dentro y lo pegamos en 000-default.conf debajo de:

|  |
| --- |
| <VirtualHost dominio:80>  ServerName dominio  ServerAlias dominio  ServerAdmin webmaster@localhost  </VirtualHost> |

|  |
| --- |
| <VirtualHost dominio:80>  </VirtualHost>  <IfModule mod\_ssl.c>  <VirtualHost \_default\_:443>  ServerAdmin webmaster@localhost  DocumentRoot /var/www/html  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  SSLEngine on  SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem  SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key  <FilesMatch "\.(cgi|shtml|phtml|php)$">  SSLOptions +StdEnvVars  </FilesMatch>  <Directory /usr/lib/cgi-bin>  SSLOptions +StdEnvVars  </Directory>  </VirtualHost>  </IfModule>  # vim: syntax=apache ts=4 sw=4 sts=4 sr noet |

## PASO 6

Le agrego el direccionamiento a django

|  |
| --- |
| <VirtualHost dominio:80>  </VirtualHost>  <IfModule mod\_ssl.c>  <VirtualHost \_default\_:443>  ServerName dominio  ServerAlias dominio  ServerAdmin webmaster@localhost  Alias /robots.txt /var/www/librosonline/static/robots.txt  Alias /favicon.ico /var/www/librosonline/static/favicon.ico  Alias /media/ /var/www/librosonline/media/  Alias /static/ /var/www/librosonline/static/  <Directory /var/www/librosonline/static>  Require all granted  </Directory>  <Directory /var/www/librosonline/media>  Require all granted  </Directory>  WSGIScriptAlias / /var/www/librosonline/librosonline/wsgi.py  <Directory /var/www/librosonline/librosonline>  <Files wsgi.py>  Require all granted  </Files>  </Directory>  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  SSLEngine on  SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem  SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key  <FilesMatch "\.(cgi|shtml|phtml|php)$">  SSLOptions +StdEnvVars  </FilesMatch>  <Directory /usr/lib/cgi-bin>  SSLOptions +StdEnvVars  </Directory>  </VirtualHost>  </IfModule>  # vim: syntax=apache ts=4 sw=4 sts=4 sr noet |

## PASO 7 – Agrego redireccionamiento

Redirecciono del http al https

Ingreso a:

|  |
| --- |
| cd /etc/apache2/sites-available |

Ingresamos a 000-default.conf y agrego un redireccionamiento:

|  |
| --- |
| Redirect / https://dominio |

Queda así:

|  |
| --- |
| <VirtualHost dominio:80>  Redirect / <https://dominio>  # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  # error, crit, alert, emerg.  # It is also possible to configure the loglevel for particular  # modules, e.g.  #LogLevel info ssl:warn  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  </VirtualHost>  <IfModule mod\_ssl.c>    </IfModule>  # vim: syntax=apache ts=4 sw=4 sts=4 sr noet |

## PASO 8 – Reinicio apache

|  |
| --- |
| systemctl restart apache2 |

## PASO 9 CERTBOT

https://certbot.eff.org/

9.1. Install snap

|  |
| --- |
| sudo apt update  sudo apt install snapd |

**9.2. Install Core**

|  |
| --- |
| sudo snap install core |

9.3. Install Certbot

|  |
| --- |
| sudo snap install --classic certbot |

9.4. Preparamos Certbot

Ejecutamos el siguiente comando para asegurarnos de que cerbot se puede ejecutar

|  |
| --- |
| sudo ln -s /snap/bin/certbot /usr/bin/certbot |

9.5. Desidir como ejecutar certbot

|  |
| --- |
| sudo certbot --apache -d dominio |

|  |
| --- |
| juanbarretor@gmail.com |

Acepto términos y condiciones y

Queremos registrar mail? n

|  |
| --- |
| Forzar que salga solo por https |

PASO 6

Ver si certbot va a renovar el certificado luego de 90 días de forma automática

|  |
| --- |
| sudo systemctl status certbot.timer |

|  |
| --- |
| You should be able to find it using this name:  snap.certbot.renew.timer |

|  |
| --- |
| Para que funcionen los javascript tienen que ser https por lo que en el uso de ajax debo cambiar este link:  <script src="http://cdn.jsdelivr.net/jquery.cookie/1.4.1/jquery.cookie.min.js"></script>  Por este:  <script src="https://cdn.jsdelivr.net/jquery.cookie/1.4.1/jquery.cookie.min.js"></script> |



### Bibliografía utilizada y sugerida

#### Libros y otros manuscritos:

Django. (2022). The web framework for perfectionists with deadlines. Django. Recuperado de https://www.djangoproject.com/.

Apache. (2022). The Apache Software Fundation. Apache. Recuperado de https://www.apache.org/.

Launch Modern Applications with NGINX. Recuperado de https://www.nginx.com/.

Hace más con tu dinero. Mercado Pago. Recuperado de https://www.mercadopago.com.ar/.

Derek DeJonghe - NGINX Cookbook, 2nd Edition - O'Reilly Media, Inc., Year: 2022.