

# Django Debug

Jorge Barón Abad

# ¿Para qué sirve?

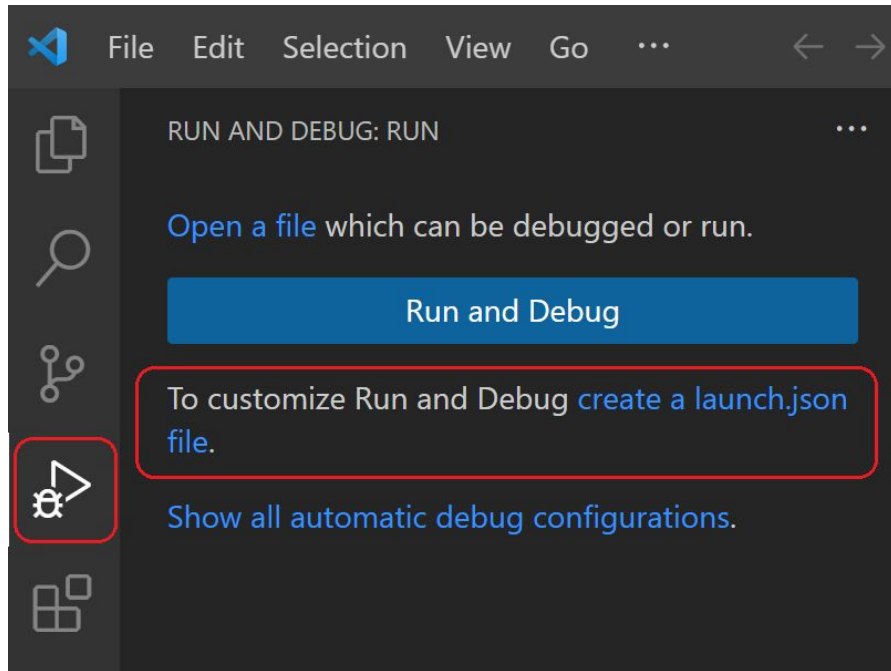
Para depurar paso a paso nuestra aplicación Django, ver información de nuestras Querys, peticiones, tiempo que tarda cada página y comprobar los posibles errores que podamos tener y corregirlos.

Explicaremos como configurar por un lado nuestra aplicación para hacer Debug en VsCode y por otro lado instalaremos la herramienta de **Debug Tool Bar**

# Configurar Debug VsCode

En VsCode, abrimos el archivo views.py pulsamos lo siguiente.

Seleccionamos Python Debugger y luego pinchamos en Django y por último manage.py



# Configurar Debug VsCode

Se creará un archivo launch.json de la siguiente forma:

```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python: Django",
      "type": "python",
      "request": "launch",
      "program": "${workspaceFolder}\\manage.py",
      "args": ["runserver"],
      "django": true,
      "justMyCode": true
    }
  ]
}
```

# Configurar Debug VsCode

En el archivo anterior debemos especificar la ruta de nuestro **manage.py**. Por ejemplo:

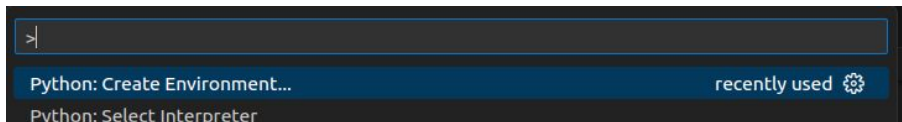
```
{
  "name": "Python: Django",
  "type": "python",
  "request": "launch",
  "program": "${workspaceFolder}/Tema_4_Views/ejercicio_vistas/manage.py"
  "args": [
    "runserver"
  ],
  "django": true,
  "justMyCode": true
},
```

# Configurar Debug VsCode

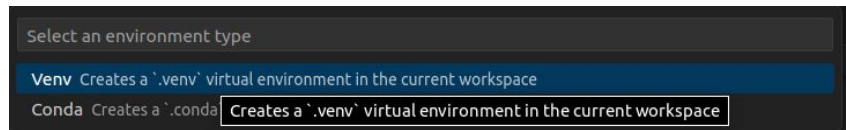
Ahora debemos crear nuestro entorno virtual de nuestro proyecto para el Debug:

Pulsamos **Ctrl+Shift+p**

Y buscamos la opción siguiente en la barra de tareas **Python:Create Enviornment**

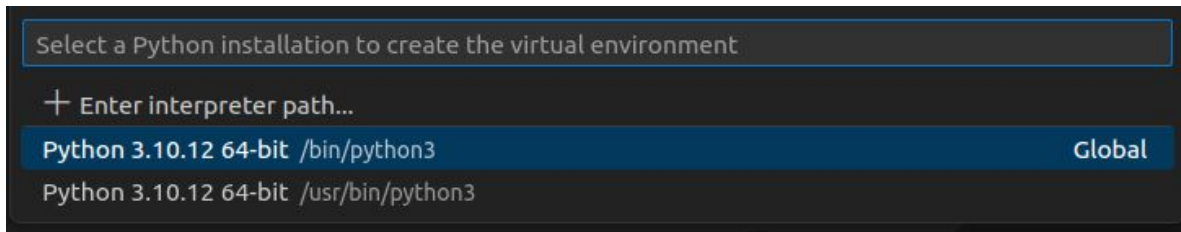


Pinchamos y ahora seleccionamos la siguiente opción: **Venv**



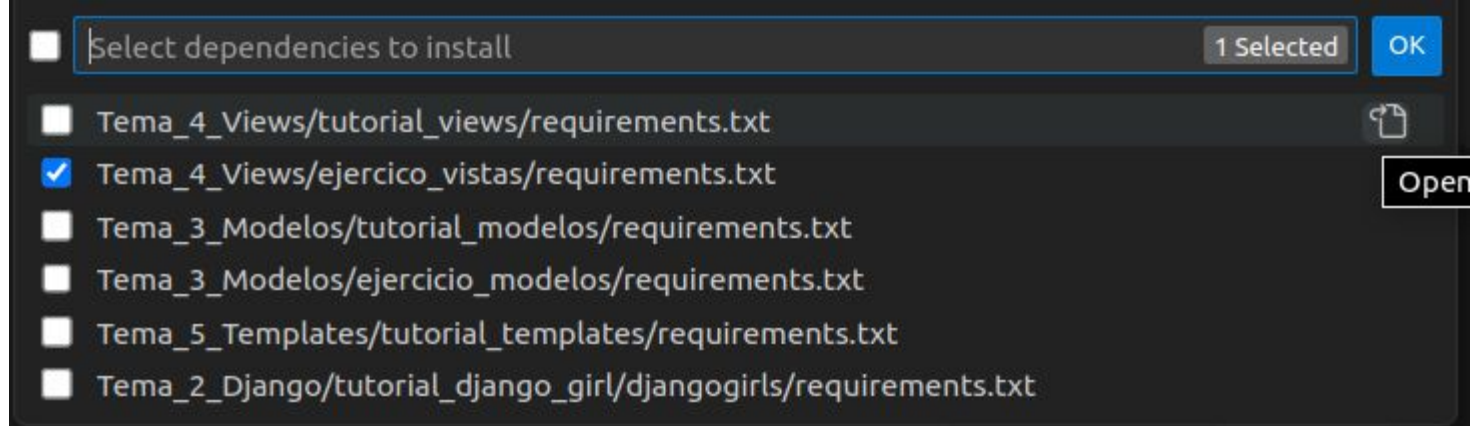
# Configurar Debug VsCode

Seleccionamos Intérprete:



# Configurar Debug VsCode

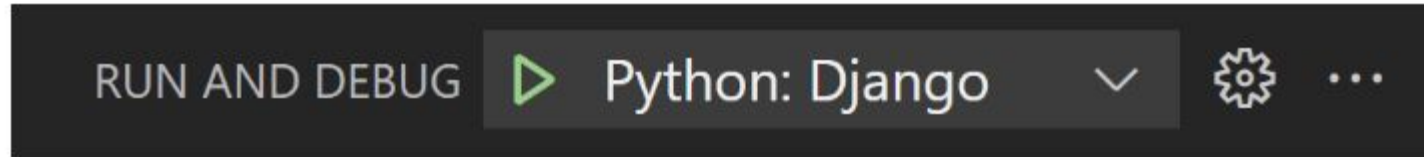
Seleccionamos la ruta del archivo de **requirements.txt** para que sepa qué requisitos tiene que instalar en el entorno:





# Configurar Debug VsCode

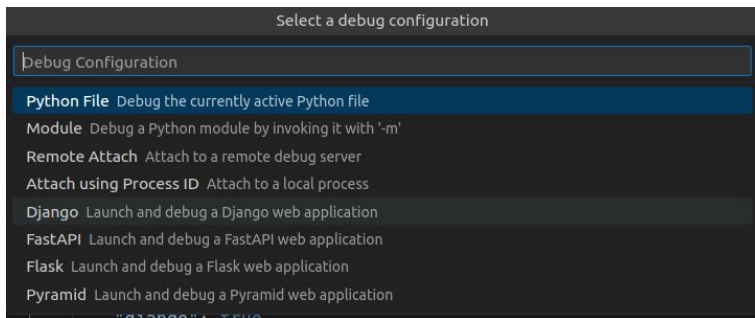
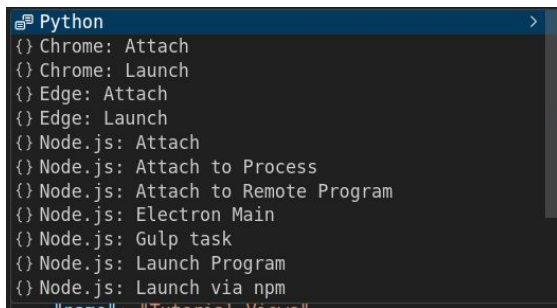
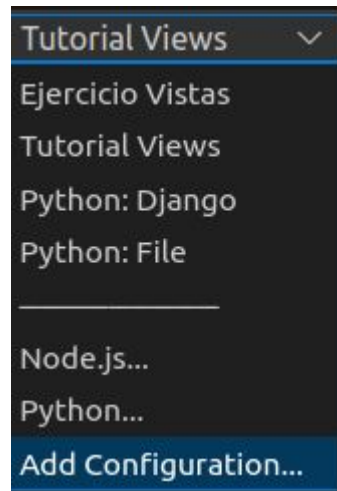
Seleccionamos nuestra configuración de Debug, que por defecto le tenemos puesto el nombre, **Python: Django** y pulsamos en ejecutar:



# Configurar Debug VsCode

Podemos añadir más configuraciones de otros proyectos, pulsando en **Add Configuration...**:

Y seleccionamos la opción de **Python** y después la de **Django**:



# Configurar Debug VsCode

Que tenemos que tener en cuenta en el archivo de configuración:

- **Name:** Nombre que le damos a esa configuración para diferenciarla de las demás y seleccionarla a la hora de depurar
- **Program:** Ruta donde se situa mi archivo **manage.py** en el proyecto
- **django:** Debe estar siempre a **True**, para que funcione con los proyectos de Django

# Debug Tool Bar

Documentación: <https://django-debug-toolbar.readthedocs.io/en/latest/index.html>

# Instalación

En el archivo requirements de nuestro proyecto incluimos:

```
django-debug-toolbar~=4.4.6
```

Después ejecutamos el comando para instalarlo

```
pip install -r requirements.txt
```

# Instalación

A continuación incluimos la aplicación **debug\_toolbar** en nuestras aplicaciones en **mysite/settings.py**:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django_seed',  
    'debug_toolbar',  
    'ejercicio'  
]
```

# Instalación

Añadimos las urls de la aplicación de debug en **mystite/urls.py**:

```
path("__debug__/", include("debug_toolbar.urls")),
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("__debug__/", include("debug_toolbar.urls")),  
    path('', include("ejercicio.urls"))  
]
```

# Instalación

Añadimos un middleware en el archivo en el archivo **mysite/settings.py**:

```
"debug_toolbar.middleware.DebugToolbarMiddleware",
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
    "debug_toolbar.middleware.DebugToolbarMiddleware",  
]
```



# Instalación

Por último en el archivo **mysite/settings.py** debemos incluir la siguiente configuración para que funcione correctamente:

```
INTERNAL_IPS = ["127.0.0.1"]
```

# Uso

Una vez que se ha realizado todos los pasos de instalación, cuando entremos en nuestra web, debería aparecer el siguiente menú lateral.

De esta forma podemos saber cuantas consultas hemos ejecutado, las peticiones que hemos realizado, el tiempo que ha tardado en cargar la página, etc..

Hide »	
History /	<input checked="" type="checkbox"/>
Versions Django 3.2.22	<input checked="" type="checkbox"/>
Time CPU: 47.93ms (47.93ms)	<input checked="" type="checkbox"/>
Settings	<input checked="" type="checkbox"/>
Headers	<input checked="" type="checkbox"/>
Request index	<input checked="" type="checkbox"/>
SQL 2 queries in 1.01ms	<input checked="" type="checkbox"/>
Static files 0 files used	<input checked="" type="checkbox"/>
Templates index.html	<input checked="" type="checkbox"/>
Cache 0 calls in 0.00ms	<input checked="" type="checkbox"/>
Signals 29 receivers of 15 signals	<input checked="" type="checkbox"/>
Intercept redirects	<input type="checkbox"/>
Profiling	<input type="checkbox"/>