



escuela técnica superior  
de ingeniería informática

# **Instalación de *Decide***

***Departamento de  
Lenguajes y Sistemas Informáticos***

**Evolución y gestión de la configuración  
4º Grado en Ingeniería Informática -  
Ingeniería del Software**

UNIVERSIDAD DE SEVILLA

# Índice

- Entorno de desarrollo vs despliegue
- Introducción a decide
- Entorno de despliegue con Docker
- Entorno de desarrollo
  - Python, virtual enviroment
- Un pequeño cambio
- Usando el cambio en el entorno de despliegue (versión 1)

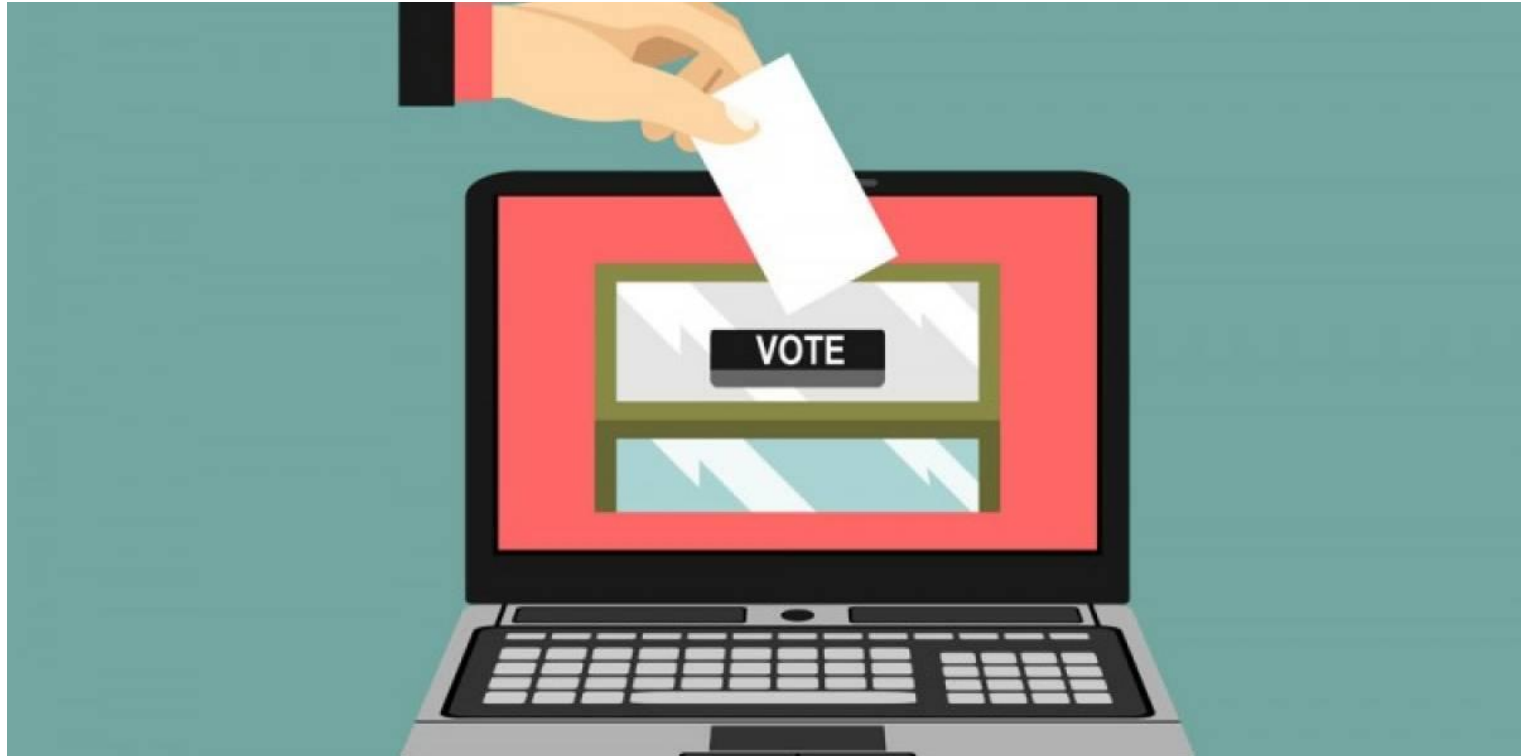


DESARROLLO



DESPLIEGUE

# La herramienta



[DECIDE →https://github.com/wadobo/decide/](https://github.com/wadobo/decide/)

[Nuestra copia de trabajo → https://github.com/EGCETSII/decide](https://github.com/EGCETSII/decide)

# Módulos

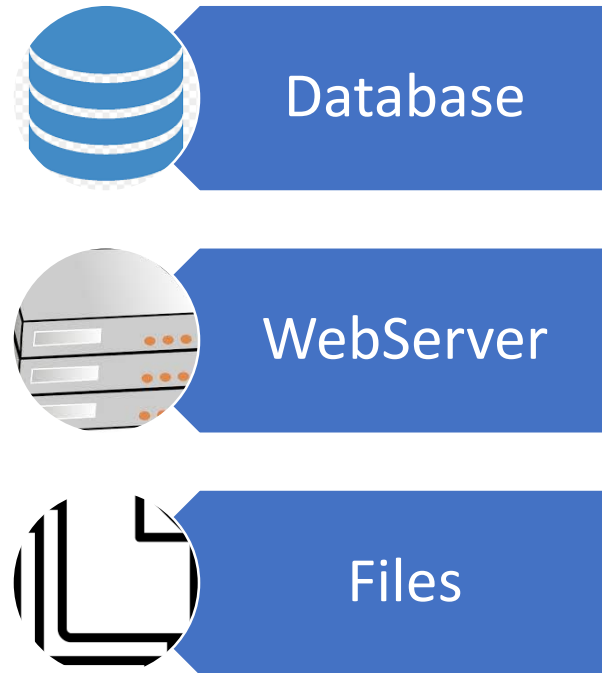
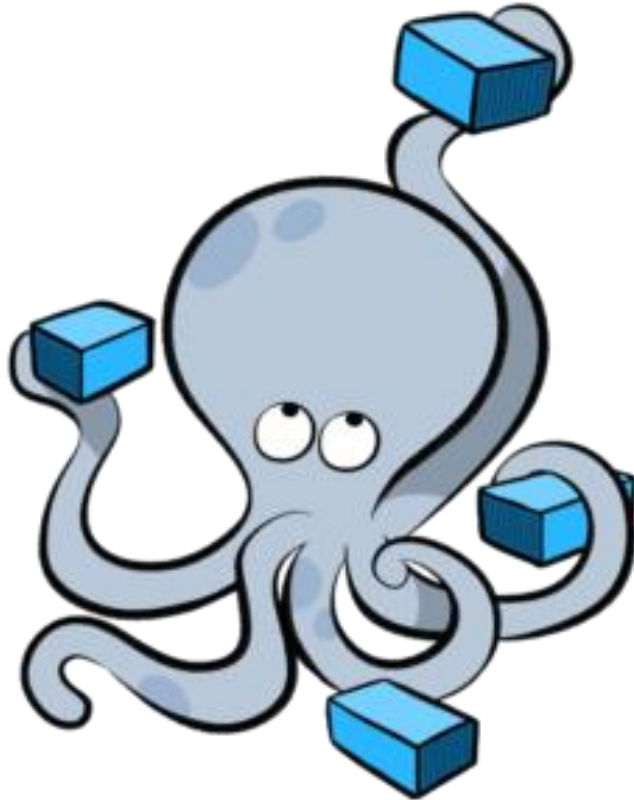
- .Authentication: Sistema de autenticación para saber quien puede votar.
- .Base: Código común o base
- .Booth: Cabina de votación.
- .Census: Censo de personas a votar.
- .Decide: Aplicación de votación (main).
- .Mixnet: Módulo de criptografía.
- .Postproc: Post-procesado de la votación.
- .Store: Almacén de votos.
- .Gateway: Enrutado de peticiones. Permite que los otros modulos se comuniquen correctamente
- .Test-scripts: Pruebas.
- .Visualizer: Visualizador de votos.
- .Voting: Votación.

**git clone <https://github.com/EGCETSII/decide.git>**

# Entorno de despliegue



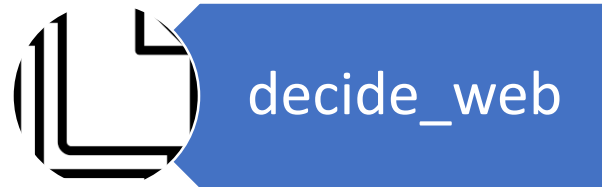
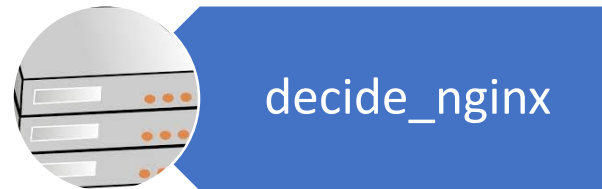
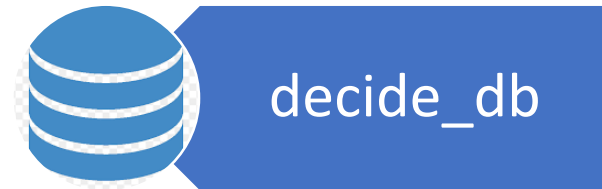
# Docker compose



Compose es una herramienta para definir y ejecutar aplicaciones Docker multicontenedores. Con Compose, se utiliza un archivo YAML para configurar los servicios de la aplicación.

# Docker compose en decide

```
services:
  db:
    restart: always
    container_name: decide_db
    image: postgres:alpine
    volumes:
      - db:/var/lib/postgresql/data
    networks:
      - decide
  web:
    restart: always
    container_name: decide_web
    image: decide_web:latest
    build: .
    command: ash -c "python manage.py migr
    expose:
      - "5000"
```



Existe una configuración de docker compose que lanza 3 contenedores, uno para el servidor de base de datos, otro para el django y otro con un servidor web nginx para servir los ficheros estáticos y hacer de proxy al servidor django.



# Ejecuta entorno despliegue docker

```
$ cd docker
```

```
$ docker-compose up -d
```

```
$ docker exec -ti decide_web ./manage.py createsuperuser
```

# Probar que todo funciona

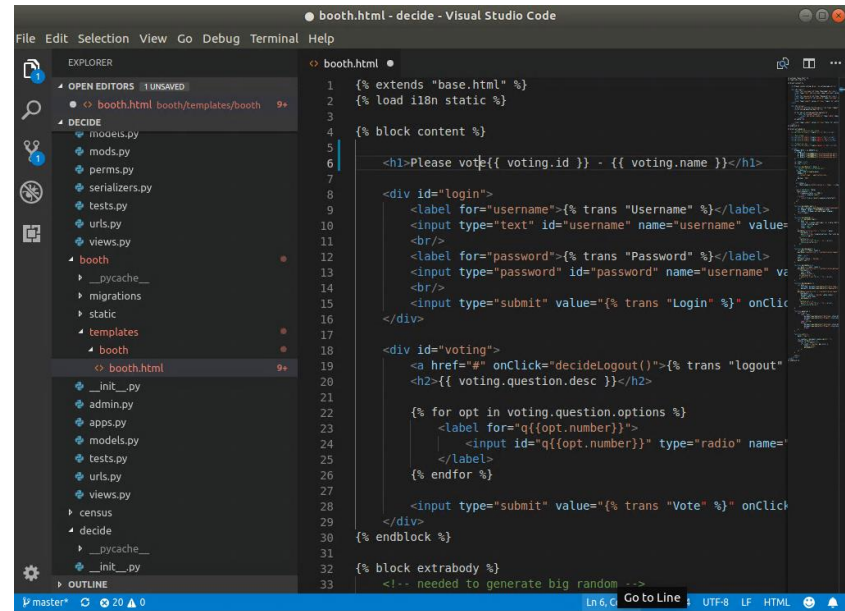
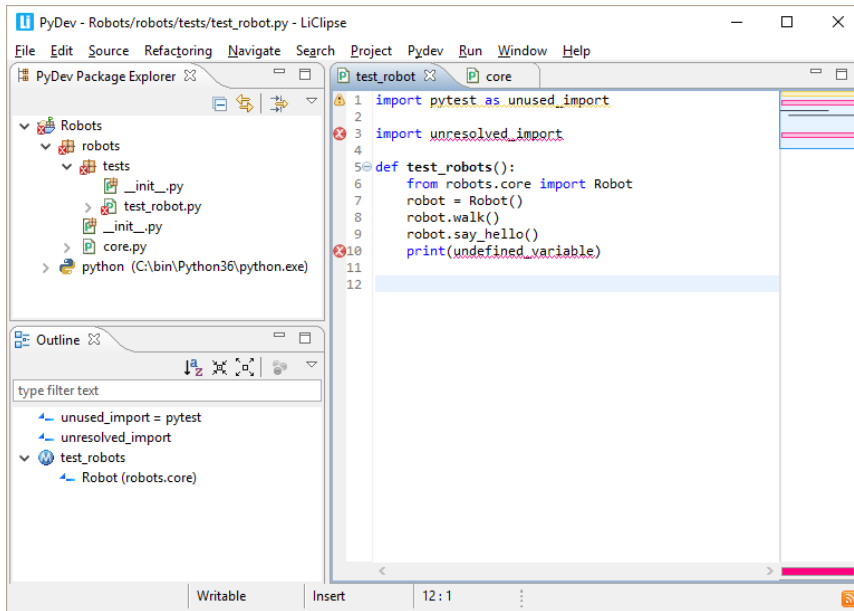
<https://github.com/wadobo/decide/wiki/Como-funciona-Decide>

# Entorno de desarrollo



# Selecciona tu IDE

- <http://www.pydev.org/>
- <https://code.visualstudio.com/>



# Entornos virtuales



## Virtualenv

```
python3 -m venv <myenvname>
```

# Toolset

Para configurar el proyecto, podremos crearnos un fichero `local_settings.py` basado en el `local_settings.example.py`, donde podremos configurar la ruta de nuestras apps o escoger que módulos ejecutar.

Una vez hecho esto, será necesario instalar las dependencias del proyecto, las cuales están en el fichero `requirements.txt`:

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

Tras esto tendremos que crearnos nuestra base de datos con postgres:

```
sudo su - postgres
```

```
psql -c "create user decide with password 'decide'"
```

```
psql -c "create database decide owner decide"
```

Entramos en la carpeta del proyecto (`cd decide`) y realizamos la primera migración para preparar la base de datos que utilizaremos:

```
./manage.py migrate
```

```
./manage.py createsuperuser
```

Por último, ya podremos ejecutar el módulos o módulos seleccionados en la configuración de la siguiente manera:

```
./manage.py runserver
```

# Localsettings.py

```
ALLOWED_HOSTS = ["*"]# Modules in use, commented modules that you won't use
MODULES = [ 'authentication', 'base', 'booth', 'census', 'mixnet', 'postproc', 'store', 'visualizer',
'voting',]
APIS = {
    'authentication': 'http://localhost:8000',
    'base': 'http://localhost:8000',
    'booth': 'http://localhost:8000',
    'census': 'http://localhost:8000',
    'mixnet': 'http://localhost:8000',
    'postproc': 'http://localhost:8000',
    'store': 'http://localhost:8000',
    'visualizer': 'http://localhost:8000',
    'voting': 'http://localhost:8000',
}
```

Mirar las URL en localhost

```
BASEURL = 'http://localhost:8000'DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'postgres',
        'USER': 'decide',
        'PASSWORD': 'decide',
        'HOST': '127.0.0.1',      'PORT': '5432',    }}# number of bits for the key, all auths should use the same number of
bitsKEYBITS = 256
```

Password de postgres

# Probando que todo funciona

Django administration

WELCOME, DE

Site administration

AUTH TOKEN

**Tokens**

[+ Add](#) [✎ Change](#)

AUTHENTICATION AND AUTHORIZATION

**Groups**

[+ Add](#) [✎ Change](#)

**Users**

[+ Add](#) [✎ Change](#)

BASE

**Auths**

[+ Add](#) [✎ Change](#)

**Keys**

[+ Add](#) [✎ Change](#)

CENSUS

**Censuss**

[+ Add](#) [✎ Change](#)

MIXNET

**Mixnets**

[+ Add](#) [✎ Change](#)

Recent actions

My actions

None available



# Haciendo un cambio

```
booth.html - decide - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

EXPLORER
OPEN EDITORS 1 UNSAVED
  booth.html booth/templates/booth 9+
DECIDE
  models.py
  mods.py
  perms.py
  serializers.py
  tests.py
  urls.py
  views.py
booth
  __pycache__
  migrations
  static
  templates
    booth
      booth.html 9+
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  census
  decide
    __pycache__
    __init__.py
  OUTLINE

booth.html
1 {% extends "base.html" %}
2 {% load i18n static %}
3
4 {% block content %}
5
6 <h1>Please vote{{ voting.id }} - {{ voting.name }}</h1>
7
8   <div id="login">
9     <label for="username">{% trans "Username" %}</label>
10    <input type="text" id="username" name="username" value="" />
11    <br/>
12    <label for="password">{% trans "Password" %}</label>
13    <input type="password" id="password" name="password" value="" />
14    <br/>
15    <input type="submit" value="{% trans "Login" %}" onClick="" />
16  </div>
17
18  <div id="voting">
19    <a href="#" onClick="decideLogout()">{% trans "logout" %}</a>
20    <h2>{{ voting.question.desc }}</h2>
21
22    {% for opt in voting.question.options %}
23      <label for="q{{opt.number}}">
24        <input id="q{{opt.number}}" type="radio" name="q{{opt.number}}" />
25        {{ opt.desc }}
26      </label>
27    {% endfor %}
28
29    <input type="submit" value="{% trans "Vote" %}" onClick="" />
30  </div>
31 </block %}
32 {% block extrabody %}
33 <!-- needed to generate big random -->
```

# Viendo el cambio

The screenshot shows a Firefox browser window with the following elements:

- System Bar:** Activities, Navegador web Firefox, and the time 'Lun 13:41'.
- Browser Tab:** 'Decide!' with a close button and a plus sign for a new tab.
- Address Bar:** 'localhost:8000/booth/1/' with navigation icons (back, forward, refresh, home) and security icons (lock, shield, star).
- Page Content:**
  - Title:** 'Please vote 1 - Votacion1' in a large, bold, black font.
  - Form:** A login form with two input fields: 'Nombre de usuario' and 'Password', followed by a 'Login' button.
- Left Panel:** A vertical dock with various application icons including Firefox, a mail client, a calendar, a help icon, Amazon, a terminal, and a code editor.
- Bottom Bar:** A status bar showing ':estado: ejecutando'.

# Probando el cambio en el entorno de despliegue

## Dockerfile

```
WORKDIR /app
```

```
RUN git clone https://github.com/wadobo/decide.git .
```

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

```
WORKDIR /app/decide
```

```
# local settings.py
```

```
ADD docker-settings.py /app/decide/local_settings.py
```



MODIFICAR PARA USAR EL  
DIRECTORIO LOCAL con  
ADD



escuela técnica superior  
de ingeniería informática

# **Instalación de *Decide***

***Departamento de  
Lenguajes y Sistemas Informáticos***

**Evolución y gestión de la configuración  
4º Grado en Ingeniería Informática -  
Ingeniería del Software**

UNIVERSIDAD DE SEVILLA