

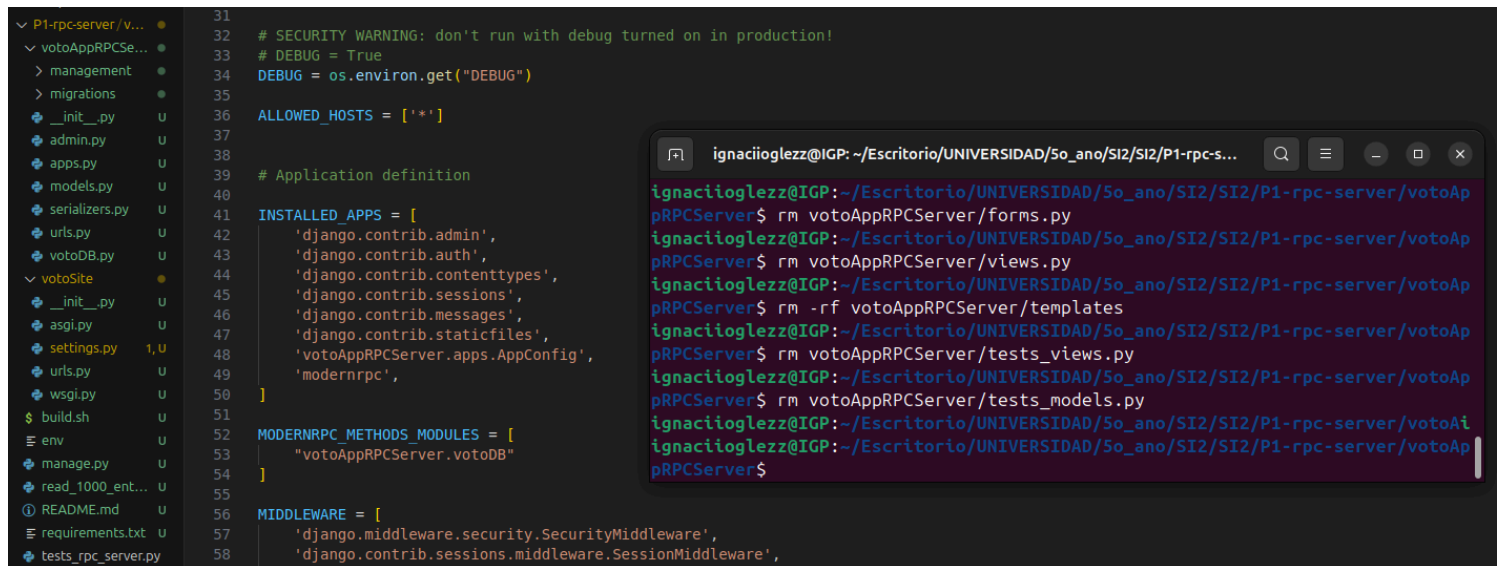
Memoria de Prácticas

Práctica 1B

Sistemas Informáticos II

Autores: Ignacio González Porras y Daniel Wentworth
Fernández

Ejercicio 1: Ejecute los pasos descritos anteriormente, uno tras otro, para crear el nuevo proyecto. Incluya en la memoria evidencias (capturas de pantalla) de haber realizado estos pasos. Indica por qué no será necesario hacer uso de los formularios Django y las plantillas en la aplicación servidor RPC.



The screenshot shows a code editor on the left with a file explorer and a terminal window on the right. The code editor displays the Django settings file for a project named 'P1-rpc-server'. The settings include security warnings, debug mode, allowed hosts, application definition, installed apps, modern RPC methods modules, and middleware. The terminal window shows a series of commands being executed to remove files from the 'votoAppRPCServer' directory.

```
31
32 # SECURITY WARNING: don't run with debug turned on in production!
33 # DEBUG = True
34 DEBUG = os.environ.get("DEBUG")
35
36 ALLOWED_HOSTS = ['*']
37
38
39 # Application definition
40
41 INSTALLED_APPS = [
42     'django.contrib.admin',
43     'django.contrib.auth',
44     'django.contrib.contenttypes',
45     'django.contrib.sessions',
46     'django.contrib.messages',
47     'django.contrib.staticfiles',
48     'votoAppRPCServer.apps.AppConfig',
49     'modernrpc',
50 ]
51
52 MODERNRPC_METHODS_MODULES = [
53     "votoAppRPCServer.votoDB"
54 ]
55
56 MIDDLEWARE = [
57     'django.middleware.security.SecurityMiddleware',
58     'django.contrib.sessions.middleware.SessionMiddleware',
```

```
ignacioglezz@IGP: ~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-s...
ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$ rm votoAppRPCServer/forms.py
ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$ rm votoAppRPCServer/views.py
ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$ rm -rf votoAppRPCServer/templates
ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$ rm votoAppRPCServer/tests_views.py
ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$ rm votoAppRPCServer/tests_models.py
ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$
```

No es necesario hacer uso de los formularios django debido a que django no se está usando como frontend, sino como un backend con procedimientos remotos. Por tanto los formularios enfocados a la vista de la aplicación no son necesarios, lo cual vemos reflejado al borrar views.py, que ya no es necesario.

Ejercicio 2: Ejecute los pasos descritos anteriormente, uno tras otro, para exportar la funcionalidad de acceso a la BD como procedimientos remotos. Incluya en la memoria evidencias (capturas de pantalla) de haber realizado estos pasos. Indica razonadamente por qué es necesario hacer uso del método `model_to_dict`.

urls.py U X

P1-rpc-server > votoAppRPCServer > votoSite > urls.py > ...

```
1 from django.urls import path
2 from modernrpc.views import RPCEntryPoint
3
4 urlpatterns = [path("rpc/", RPCEntryPoint.as_view(), name="rpc")]
```

```

def registrar_voto(voto_dict):
    try:
        voto = Voto.objects.create(**voto_dict)
        # get default values from voto
        voto = Voto.objects.get(pk=voto.pk)
    except Exception as e:
        print("Error: Registrando voto: ", e)
        return None

    voto_a_devolver = model_to_dict(voto)
    voto_a_devolver['marcaTiempo'] = str(voto_a_devolver['marcaTiempo'])
    return voto_a_devolver


@rpc_method
def eliminar_voto(idVoto):
    """ Delete a vote in the database
    :param idVoto: id of the vote to be deleted
    :return True if succesful,
    False otherwise
    """
    try:
        voto = Voto.objects.get(id=idVoto)
    except Voto.DoesNotExist:
        return False
    voto.delete()
    return True


@rpc_method
def get_votos_from_db(idProcesoElectoral):
    """ Gets votes in the database correspondint to some electoral processs
    :param idProcesoElectoral: id of the vote to be deleted
    :return list of votes found
    """

    votos_a_devolver = []

    votos = Voto.objects.filter(idProcesoElectoral=idProcesoElectoral)

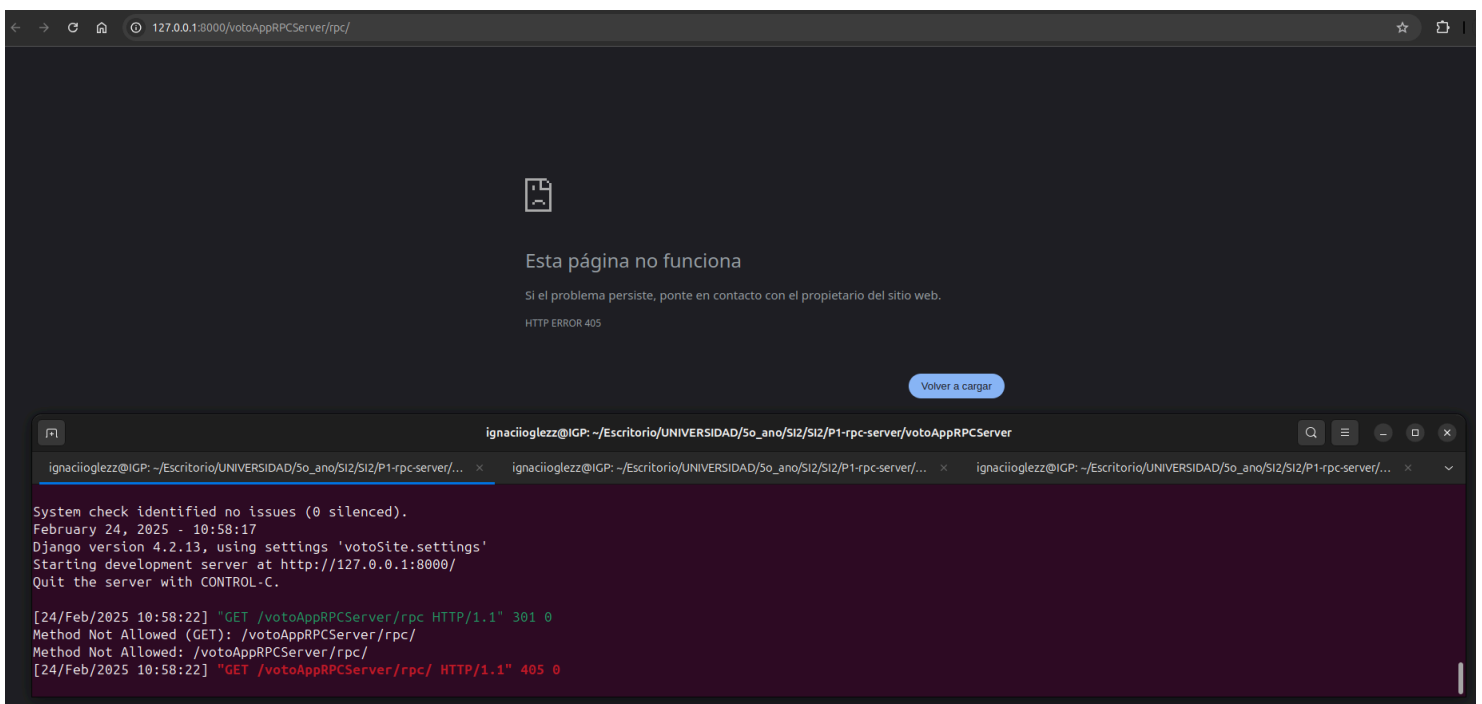
    for voto in votos:
        voto_dict = model_to_dict(voto)
        voto_dict['marcaTiempo'] = str(voto_dict['marcaTiempo'])
        votos_a_devolver.append(voto_dict)

    return votos_a_devolver

```

Es necesario hacer uso del método `model_to_dict` para poder modificar los valores que nos interesen de las instancias de los modelos.

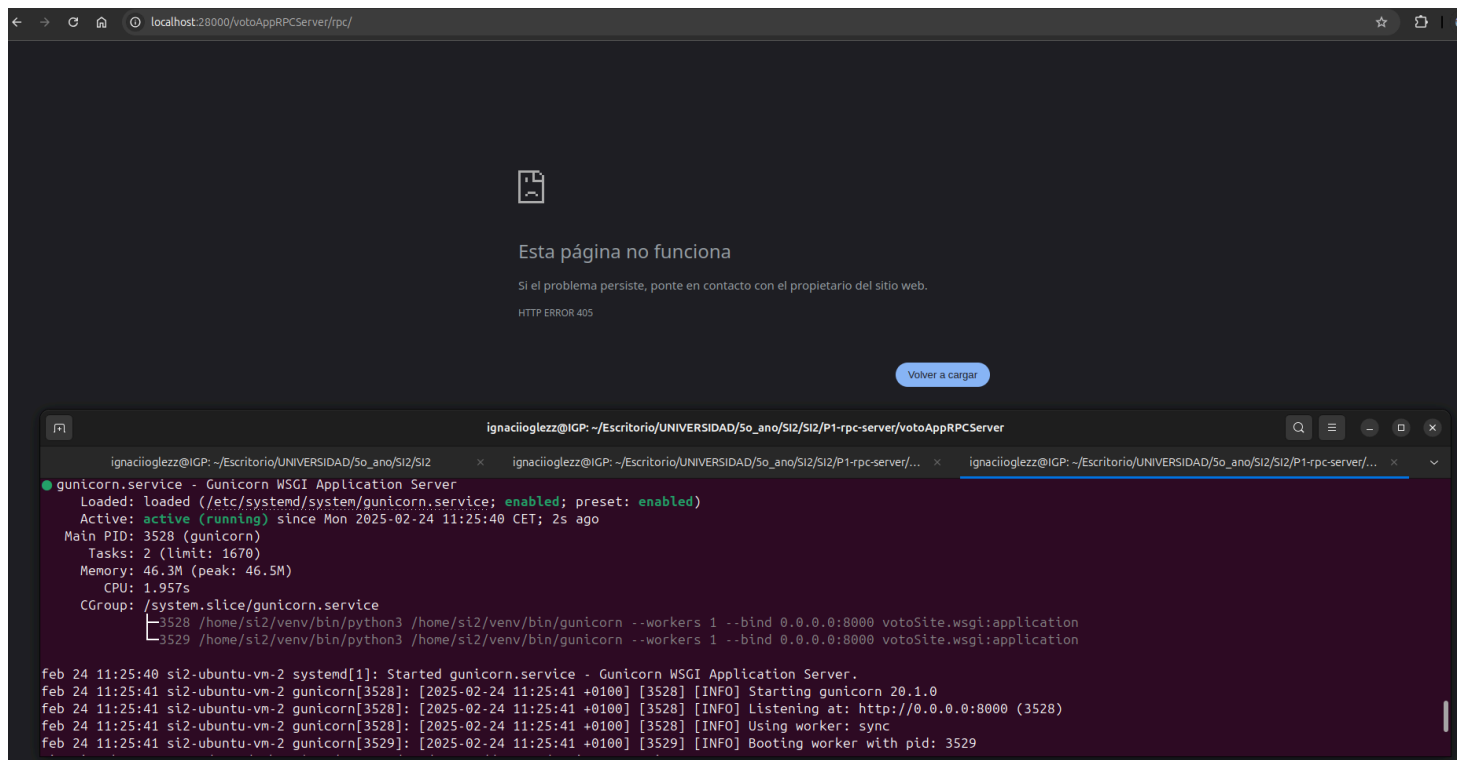
Ejercicio 3: Ejecuta los pasos descritos más arriba e incluye evidencias en la memoria de haberlos llevado a cabo. Ejecuta los test proporcionados con el proyecto y comprueba que no devuelven errores. Adjunta en la memoria una captura de pantalla en la que se muestre el resultado de ejecutar los test (`python manage.py test votoAppRPCServer.tests_rpc_server`). Los test proporcionados deben tomarse como requisitos extras del sistema.



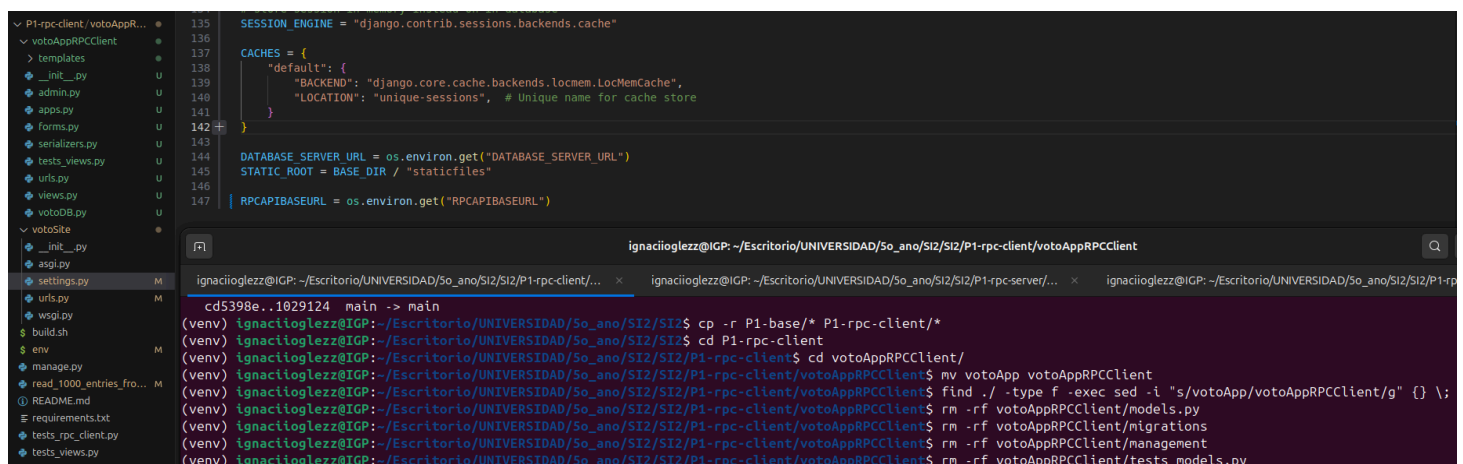
```
(venv) ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-server/votoAppRPCServer$ python manage.py test
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 4 tests in 0.075s

OK
Destroying test database for alias 'default'...
```

Ejercicio 4: Ejecuta los pasos descritos más arriba. Prueba a acceder desde el navegador del PC del laboratorio a la URL `localhost:28000/votoAppRPCServer/rpc` para comprobar que el despliegue de la aplicación es correcto. Incluye evidencias en la memoria en forma de capturas de pantalla mostrando que el acceso a dicha URL es correcto.



Ejercicio 5: Ejecute los pasos descritos anteriormente, uno tras otro, para crear el nuevo proyecto. Incluya en la memoria evidencias (capturas de pantalla) de haber realizado estos pasos. Indica por qué no será necesario hacer uso de los modelos de datos Django en la aplicación cliente RPC.



Ejercicio 6: Ejecute los pasos descritos anteriormente, uno tras otro, para crear el nuevo fichero votoDB.py, que invoque la funcionalidad de la aplicación como llamadas a procedimiento remoto. Comenta cada función describiendo tanto los argumentos de entrada como los valores devueltos. Incluya en la memoria evidencias (capturas de pantalla) de haber realizado estos pasos. Indica qué tipo de binding, de los tipos vistos en las transparencias de teoría sobre RPC, realiza el cliente RPC codificado.

127.0.0.1:8000/votoAppRPCClient/

Introduzca la Informacion Censo de la Persona que Vota (votoSite)

Número de DNI:

Nombre:

Fecha de Nacimiento:

Código de Autorización:

Enviar Información Censo

ignacioglezz@IGP: ~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient

OK
Destroying test database for alias 'default'...
(venv) ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient\$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 24, 2025 - 12:18:36
Django version 4.2.13, using settings 'votoSite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

[24/Feb/2025 12:18:39] "GET /votoAppRPCClient/ HTTP/1.1" 200 991

127.0.0.1:8000/votoAppRPCClient/testbd/

Test Base de Datos: Registro de Voto (votoSite)

Introduzca los datos del nuevo voto a registrar:

ID Proceso Electoral:

ID Circunscripción:

ID Mesa Electoral:

Nombre Candidato Votado:

Número de DNI:

Nombre:

Fecha de Nacimiento:

Código de Autorización:

Registrar Voto

Test Base de Datos: Borrado de Voto

Introduzca el ID del voto a Borrar:

ID del Voto:

Borrar Voto

Test Base de Datos: Listado de Votos

Introduzca el ID del proceso electoral:

ID del Proceso Electoral:

Obtener Votos

ignacioglezz@IGP: ~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient

(venv) ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient\$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 24, 2025 - 12:18:36
Django version 4.2.13, using settings 'votoSite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

[24/Feb/2025 12:18:39] "GET /votoAppRPCClient/ HTTP/1.1" 200 991
[24/Feb/2025 12:20:08] "GET /votoAppRPCClient/voto HTTP/1.1" 301 0
[24/Feb/2025 12:20:08] "GET /votoAppRPCClient/voto/ HTTP/1.1" 200 1081
[24/Feb/2025 12:20:39] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

127.0.0.1:8000/votoAppRPCClient/voto/

Registro de Voto (votoSite)

Introduzca los datos del nuevo voto a registrar:

ID Proceso Electoral:

ID Circunscripción:

ID Mesa Electoral:

Nombre Candidato Votado:

Enviar Información Voto

ignacioglezz@IGP: ~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient

Destroying test database for alias 'default'...
(venv) ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/So_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient\$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 24, 2025 - 12:18:36
Django version 4.2.13, using settings 'votoSite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

[24/Feb/2025 12:18:39] "GET /votoAppRPCClient/ HTTP/1.1" 200 991
[24/Feb/2025 12:20:08] "GET /votoAppRPCClient/voto HTTP/1.1" 301 0
[24/Feb/2025 12:20:08] "GET /votoAppRPCClient/voto/ HTTP/1.1" 200 1081

```
def verificar_censo(censo_data):
    """ Check if the voter is registered in the Censo
    :param censo_dict: dictionary with the voter data
    | (as provided by CensoForm)
    :return True or False if censo_data is not valid
    """
    if bool(censo_data) is False or not\
        with ServerProxy(settings.RPCAPIBASEURL) as proxy:
            return proxy.verificar_censo(censo_data)

    return False

def registrar_voto(voto_dict):
    """ Register a vote in the database
    :param voto_dict: dictionary with the vote data (as provided by VotoForm)
    | plus de censo_id (numeroDNI) of the voter
    :return new voto info if succesful, None otherwise
    """
    try:
        with ServerProxy(settings.RPCAPIBASEURL) as proxy:
            voto = proxy.registrar_voto(voto_dict)

    except Exception as e:
        print("Error: Registrando voto: ", e)
        return None

    return voto

def eliminar_voto(idVoto):
    """ Delete a vote in the database
    :param idVoto: id of the vote to be deleted
    :return True if succesful,
    | False otherwise
    """
    try:
        with ServerProxy(settings.RPCAPIBASEURL) as proxy:
            return proxy.eliminar_voto(idVoto)

    except Exception as e:
        print("Error: Eliminando voto: ", e)
        return False

    return True
```

Se trata de un binding estático debido a que el cliente debe conocer la ubicación del servidor de antemano, que está definida en compilación.

Ejercicio 7: Ejecuta los pasos descritos más arriba e incluye evidencias en la memoria de haberlos llevado a cabo. Ejecuta los test proporcionados con el proyecto sobre las vistas y comprueba que no devuelven errores. Adjunta en la memoria una captura de pantalla en la que se muestre el resultado de ejecutar los test (python manage.py test).

```
$ env M x
P1-rpc-client > votoAppRPCClient > $ env
1 # IMPORTANT: this file should not be in a repository
2 # To remove a file named env from a Git repository
3 # but keep it in the source (local system), follow these steps:
4 # Remove the file from Git tracking but keep it locally
5 ## git rm --cached env
6 # Add 'env' to .gitignore (so it's not tracked again)
7 ## echo "env" >> .gitignore
8 # Commit the changes
9 ## git commit -m "Removed env from Git tracking and added to .gitignore"
10 # Push the changes to the remote repository
11 ## git push
12 # use sqlite 3
13 ## DATABASE_SERVER_URL=sqlite:///db.sqlite3
14 # use postgres
15 DATABASE_SERVER_URL='postgres://alumnodb:alumnodb@localhost:15432/voto'
16 DATABASE_SERVER_URL='postgres://neondb_owner:npg_pxukNK7EDw4Q@ep-curly-hall-a9z21sjf
17 # The client does not need to store data in any database
18 # so let us define a sqlite in order to avoid warning messages
19 DEBUG=True
20 SECRET_KEY = 'django-insecure-alczftnjl#$v%xmK@5j(n*px43c8kxgl_ua4%khc+t7g_)s9d'
21 RPCAPIBASEURL='http://localhost:28000/votoAppRPCServer/rpc/'
22
23
settings.py x
P1-rpc-client > votoAppRPCClient > votoSite > settings.py > ...
129 # Default primary key field type
130 # https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field
131
132 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
133
134 # store session in memory instead on in database
135 SESSION_ENGINE = "django.contrib.sessions.backends.cache"
136
137 CACHES = {
138     "default": {
139         "BACKEND": "django.core.cache.backends.locmem.LocMemCache",
140         "LOCATION": "unique-sessions", # Unique name for cache store
141     }
142 }
143
144 DATABASE_SERVER_URL = os.environ.get("DATABASE_SERVER_URL")
145 STATIC_ROOT = BASE_DIR / "staticfiles"
146
147 RPCAPIBASEURL = os.environ.get("RPCAPIBASEURL")

votoDB.py x
P1-rpc-client > votoAppRPCClient > votoAppRPCClient > votoDB.py > ...
1 #
2 # it under the terms of the GNU General Public License as published by
3 # the Free Software Foundation, either version 3 of the License, or
4 # (at your option) any later version.
5 #
6 # author: rmarabini
7 "Interface with the databse"
8 from django.conf import settings
9 from xmlrpc.client import ServerProxy
10
11
12 def verificar_censo(censo_data):
13     """ Check if the voter is registered in the Censo
14     :param censo_dict: dictionary with the voter data
15                       (as provided by CensoForm)
16     :return True or False if censo_data is not valid
17     """
18     if not censo_data:
19         return False
20
21     with ServerProxy(settings.RPCAPIBASEURL) as proxy:
22         return proxy.verificar_censo(censo_data)
23
24
```

```
(venv) ignacioglezz@IGP:~/Escritorio/UNIVERSIDAD/5o_ano/SI2/SI2/P1-rpc-client/votoAppRPCClient$ python manage.py test
Found 17 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..voto {'id': 12, 'idCircunscripcion': 'CIRC123', 'idMesaElectoral': 'MESA123', 'idProcesoElectoral': 'ELEC123', 'nombreCandidatoVotado': 'Candid
ate A', 'censo': '23', 'codigoRespuesta': '000', 'marcaTiempo': '2025-02-24 11:22:37.377168+00:00'}
.....num_rows: 1
voto_id: 19
.....
Ran 17 tests in 0.571s

OK
Destroying test database for alias 'default'...
```

Ejercicio 8: Ejecuta los pasos descritos más arriba. Prueba a acceder desde el navegador del PC del laboratorio a la URL localhost:38000/votoAppRPCClient/ para comprobar que el despliegue de la aplicación es correcto. Registra un voto, lístalo (tanto desde la aplicación usando testbd como desde un cliente SQL como DBeaver), y bórralo. Registra el voto tanto desde localhost:38000/votoAppRPCClient/ como desde localhost:38000/votoAppRPCClient/testbd. Incluye evidencias en la memoria en forma de capturas de pantalla.

← → ↻ 🏠 ⓘ 127.0.0.1:38000/votoAppRPCClient/voto/

Registro de Voto (votoSite)

Introduzca los datos del nuevo voto a registrar:

ID Proceso Electoral:

ID Circunscripción:

ID Mesa Electoral:

Nombre Candidato Votado:

Enviar Información Voto

ignacioglezz@IGP: ~

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~

Django version 4.2.13, using settings 'votoSite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

^Csi2@si2-ubuntu-vm-3:~/repo/pibase/P1-rpc-client/votoAppRPCClient\$ python manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 24, 2025 - 12:38:47
Django version 4.2.13, using settings 'votoSite.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.

[24/Feb/2025 12:38:50] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495
Not Found: /favicon.ico
[24/Feb/2025 12:38:50] "GET /favicon.ico HTTP/1.1" 404 2230
[24/Feb/2025 12:39:03] "GET /votoAppRPCClient/ HTTP/1.1" 200 991
[24/Feb/2025 12:39:33] "POST /votoAppRPCClient/ HTTP/1.1" 302 0
[24/Feb/2025 12:39:33] "GET /votoAppRPCClient/voto/ HTTP/1.1" 200 1081

← → ↻ 🏠 ⓘ 127.0.0.1:38000/votoAppRPCClient/testbd/getvotos/

Votos Registrados (votoSite)

| id | idCircunscripcion | idMesaElectoral | Candidato Votado | Marca Tiempo | Codigo Respuesta |
|----|-------------------|-----------------|------------------|----------------------------------|------------------|
| 24 | 1 | 1 | 1 | 2025-02-24 11:43:42.558990+00:00 | 000 |
| 25 | 6 | 5 | 10 | 2025-02-24 11:44:11.183955+00:00 | 000 |

ignacioglezz@IGP: ~

voto=# select * from voto;

| id | idCircunscripcion | idMesaElectoral | idProcesoElectoral | nombreCandidatoVotado | marcaTiempo | codigoRespuesta | censo_id |
|----|-------------------|-----------------|--------------------|-----------------------|-------------------------------|-----------------|-----------|
| 24 | 1 | 1 | 1 | 1 | 2025-02-24 12:43:42.55899+01 | 000 | 39739740E |
| 25 | 6 | 5 | 1 | 10 | 2025-02-24 12:44:11.183955+01 | 000 | 83583583L |
| 26 | 2 | 8 | 5 | 9 | 2025-02-24 12:44:48.674283+01 | 000 | 38138138W |

(3 rows)

voto=#

127.0.0.1:38000/votoAppRPCClient/testbd/delvoto/

¡Voto eliminado correctamente!

ignacioglezz@IGP: ~

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~

[24/Feb/2025 12:38:50] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

Not Found: /favicon.ico

[24/Feb/2025 12:38:50] "GET /favicon.ico HTTP/1.1" 404 2230

[24/Feb/2025 12:39:03] "GET /votoAppRPCClient/ HTTP/1.1" 200 991

[24/Feb/2025 12:39:33] "POST /votoAppRPCClient/ HTTP/1.1" 302 0

[24/Feb/2025 12:39:33] "GET /votoAppRPCClient/voto/ HTTP/1.1" 200 1081

[24/Feb/2025 12:40:25] "POST /votoAppRPCClient/voto/ HTTP/1.1" 200 407

Not Found: /votoAppRPCClient/testbd/

[24/Feb/2025 12:41:49] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 404 3274

[24/Feb/2025 12:41:54] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

[24/Feb/2025 12:42:10] "POST /votoAppRPCClient/testbd/delvoto/ HTTP/1.1" 200 186

ignacioglezz@IGP: ~

postgres=# \c voto

You are now connected to database "voto" as user "postgres".

voto=# select * from voto;

| id | idCircunscripcion | idMesaElectoral | idProcesoElectoral | nombreCandidatoVotado | marcaTiempo | codigoRespuesta | censo_id |
|----|-------------------|-----------------|--------------------|-----------------------|-------------------------------|-----------------|-----------|
| 23 | 1 | 1 | 1 | | 2025-02-24 12:40:25.958262+01 | 000 | 60060060C |

(1 row)

voto=# select * from voto;

| id | idCircunscripcion | idMesaElectoral | idProcesoElectoral | nombreCandidatoVotado | marcaTiempo | codigoRespuesta | censo_id |
|----|-------------------|-----------------|--------------------|-----------------------|-------------|-----------------|----------|
|----|-------------------|-----------------|--------------------|-----------------------|-------------|-----------------|----------|

(0 rows)

voto=#

127.0.0.1:38000/votoAppRPCClient/voto/

Voto Registrado con Éxito (votoSite)

Id: 23

Codigo Respuesta: 000

Marca Tiempo : 2025-02-24 11:40:25.958262+00:00

Id Circunscripcion : 1

Id Mesa Electoral : 1

Id Proceso Electoral : 1

Nombre Candidato Votado: 1

ignacioglezz@IGP: ~

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~

February 24, 2025 - 12:38:47

Django version 4.2.13, using settings 'votoSite.settings'

Starting development server at http://0.0.0.0:8000/

Quit the server with CONTROL-C.

[24/Feb/2025 12:38:50] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

Not Found: /favicon.ico

[24/Feb/2025 12:38:50] "GET /favicon.ico HTTP/1.1" 404 2230

[24/Feb/2025 12:39:03] "GET /votoAppRPCClient/ HTTP/1.1" 200 991

[24/Feb/2025 12:39:33] "POST /votoAppRPCClient/ HTTP/1.1" 302 0

[24/Feb/2025 12:39:33] "GET /votoAppRPCClient/voto/ HTTP/1.1" 200 1081

[24/Feb/2025 12:40:25] "POST /votoAppRPCClient/voto/ HTTP/1.1" 200 407

ignacioglezz@IGP: ~

Last login: Mon Feb 24 10:54:10 2025 from 10.0.2.2

si2@si2-ubuntu-vm-2:~\$ sudo -u postgres psql

psql (16.6 (Ubuntu 16.6-0ubuntu0.24.04.1))

Type "help" for help.

postgres=# \c voto

You are now connected to database "voto" as user "postgres".

voto=# select * from voto;

| id | idCircunscripcion | idMesaElectoral | idProcesoElectoral | nombreCandidatoVotado | marcaTiempo | codigoRespuesta | censo_id |
|----|-------------------|-----------------|--------------------|-----------------------|-------------------------------|-----------------|-----------|
| 23 | 1 | 1 | 1 | | 2025-02-24 12:40:25.958262+01 | 000 | 60060060C |

(1 row)

voto=#

127.0.0.1:38000/votoAppRPCClient/testbd/

Voto Registrado con Éxito (votoSite)

Id: 27

Codigo Respuesta: 000

Marca Tiempo : 2025-02-24 11:50:24.827809+00:00

Id Circunscripcion : 2

Id Mesa Electoral : 8

Id Proceso Electoral : 5

Nombre Candidato Votado: 9

ignacioglezz@IGP: ~

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~/Escrit...

ignacioglezz@IGP: ~

[24/Feb/2025 12:41:54] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

[24/Feb/2025 12:42:10] "POST /votoAppRPCClient/testbd/delvoto/ HTTP/1.1" 200 186

[24/Feb/2025 12:42:51] "POST /votoAppRPCClient/testbd/ HTTP/1.1" 200 198

[24/Feb/2025 12:42:53] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

[24/Feb/2025 12:43:42] "POST /votoAppRPCClient/testbd/ HTTP/1.1" 200 407

[24/Feb/2025 12:43:45] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

[24/Feb/2025 12:44:11] "POST /votoAppRPCClient/testbd/ HTTP/1.1" 200 408

[24/Feb/2025 12:44:13] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

[24/Feb/2025 12:44:48] "POST /votoAppRPCClient/testbd/ HTTP/1.1" 200 407

[24/Feb/2025 12:44:56] "GET /votoAppRPCClient/testbd/ HTTP/1.1" 200 2495

[24/Feb/2025 12:45:08] "POST /votoAppRPCClient/testbd/getvotos/ HTTP/1.1" 200 659

[24/Feb/2025 12:50:24] "POST /votoAppRPCClient/testbd/ HTTP/1.1" 200 407

ignacioglezz@IGP: ~

| id | idCircunscripcion | idMesaElectoral | idProcesoElectoral | nombreCandidatoVotado | marcaTiempo | codigoRespuesta | censo_id |
|----|-------------------|-----------------|--------------------|-----------------------|-------------------------------|-----------------|-----------|
| 24 | 1 | 1 | 1 | | 2025-02-24 12:43:42.55899+01 | 000 | 39739740E |
| 25 | 6 | 5 | 1 | 10 | 2025-02-24 12:44:11.183955+01 | 000 | 83583583L |
| 26 | 2 | 8 | 5 | 9 | 2025-02-24 12:44:48.674283+01 | 000 | 38138138W |
| 27 | 2 | 8 | 5 | 9 | 2025-02-24 12:50:24.827809+01 | 000 | 60060060C |

(4 rows)

voto=#

Ejercicio 9: Ejecuta los pasos descritos más arriba. Incluye evidencias en la memoria en forma de capturas de pantalla del código desarrollado.

```
server_mq.py X
P1-rpc-server > votoAppRPCServer > votoAppRPCServer > server_mq.py > ...
16 def main():
17
18     print("Debe indicar el host y el puerto")
19     exit()
20
21
22     hostname = sys.argv[1]
23     port = sys.argv[2]
24
25     # TODO: completar segun las indicaciones
26
27     connection = pika.BlockingConnection(pika.ConnectionParameters(host=hostname, port=port, credentials=pika.PlainCredentials('alumnomq', 'alumnomq')))
28     channel = connection.channel()
29
30     channel.queue_declare(queue='voto_cancelacion')
31
32     def callback(ch, method, properties, body):
33         try:
34             # Procesar el mensaje
35             voto_id = body.decode('utf-8')
36             print(f" [X] Recibida solicitud para cancelar voto con ID: {voto_id}")
37
38             # Buscar el voto y cambiar su código a '111' para marcarlo como cancelado
39             try:
40                 voto = Voto.objects.get(id=voto_id)
41                 voto.codigo_respuesta = '111'
42                 voto.save()
43                 print(f" [X] Voto con ID {voto_id} cancelado correctamente")
44             except Voto.DoesNotExist:
45                 print(f" [X] Voto con ID {voto_id} no encontrado, cancelación fallida")
46             except Exception as e:
47                 print(f" [X] Error al cancelar el voto: {e}")
48
49             # Confirmar el mensaje
50             ch.basic_ack(delivery_tag=method.delivery_tag)
51         except Exception as e:
52             print(f" [X] Error al procesar el mensaje: {e}")
53
54     # Establecer QoS para procesar mensajes uno a uno
55     channel.basic_qos(prefetch_count=1)
56
57     # Asociar la función callback a la cola
58     channel.basic_consume(queue='voto_cancelacion', on_message_callback=callback)
59
60     print(' [*] Esperando mensajes. Para salir presione CTRL+C')
61     channel.start_consuming()
62
63 if __name__ == "__main__":
64
65     main()
```

Ejercicio 10: Ejecuta los pasos descritos más arriba. Incluye evidencias en la memoria en forma de capturas de pantalla del código desarrollado.

```
client_mq.py x
P1-rpc-client > votoAppRPCClient > cliente_mom > client_mq.py > ...
1 import pika
2 import sys
3
4 def cancelar_voto(hostname, port, id_voto):
5
6     try:
7         # TODO: conectar con rabbitMQ
8         connection = pika.BlockingConnection(pika.ConnectionParameters(host=hostname, port=port, credentials=pika.PlainCredentials('alumnomq', 'alumnomq')))
9     except Exception as e:
10         print("Error al conectar al host remoto:", e)
11         exit()
12
13     # TODO: declarar cola y publicar mensaje
14     channel = connection.channel()
15
16     channel.queue_declare(queue='voto_cancelacion')
17
18     channel.basic_publish(exchange='', routing_key='voto_cancelacion', body=id_voto)
19
20     print(f"[X] Enviada solicitud de cancelación para el voto ID: {id_voto}")
21
22     # TODO: cerrar conexion
23     connection.close()
24
25 def main():
26
27     if len(sys.argv) != 4:
28         print("Debe indicar el host, el numero de puerto, y el ID del voto a cancelar como un argumento.")
29         exit()
30
31     cancelar_voto(sys.argv[1], sys.argv[2], sys.argv[3])
32
33 if __name__ == "__main__":
34     main()
```

Ejercicio 11: En este ejercicio probaremos a ejecutar el código desarrollado en los pasos anteriores.

1)

```
(venv) ignactioglezz@ignactioglezz:~/Escritorio/UNIVERSIDAD/So_año/SI2/SI2/P1-rpc-client/votoAppRPCClient/cliente_mom$ python client_mq.py 192.168.86.35 15672 35
[X] Enviada solicitud de cancelación para el voto ID: 35
```

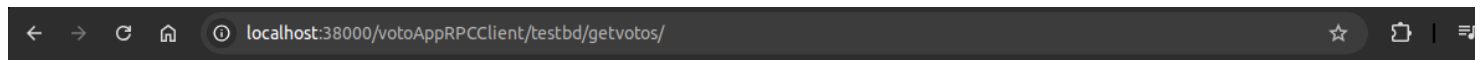
2)

```
si2@si2-ubuntu-vm-3:~$ sudo rabbitmqctl list_queues
Timeout: 60.0 seconds ...
Listing queues for vhost / ...
name      messages
voto_cancelacion 1
```

3)

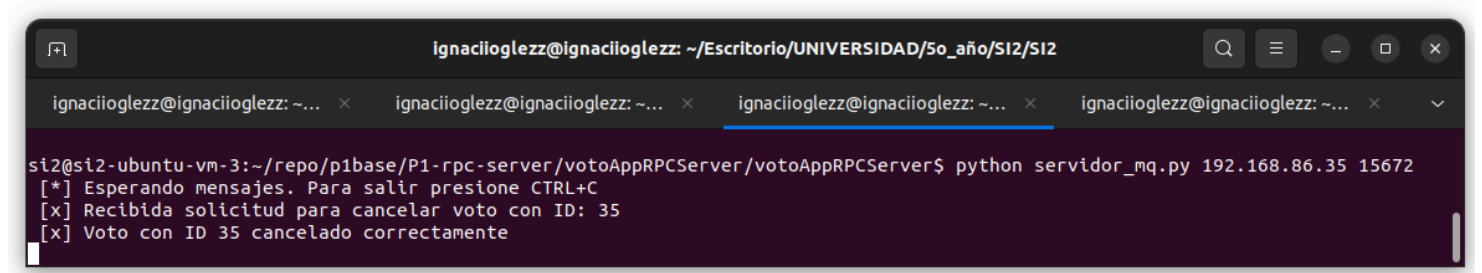
```
si2@si2-ubuntu-vm-3:~/repo/pibase/P1-rpc-server/votoAppRPCServer/votoAppRPCServer$ python servidor_mq.py 192.168.86.35 15672
[*] Esperando mensajes. Para salir presione CTRL+C
[X] Recibida solicitud para cancelar voto con ID: 35
[X] Voto con ID 35 cancelado correctamente
```

4)



Votos Registrados (votoSite)

| id | IdCircunscripcion | IdMesaElectoral | Candidato Votado | Marca Tiempo | Codigo Respuesta |
|----|-------------------|-----------------|------------------|----------------------------------|------------------|
| 34 | 3 | 3 | 3 | 2025-03-07 10:43:31.794436+00:00 | 000 |
| 36 | 5 | 5 | 5 | 2025-03-07 10:44:21.523787+00:00 | 000 |
| 35 | 4 | 4 | 4 | 2025-03-07 11:00:38.669398+00:00 | 111 |



5)

```
si2@si2-ubuntu-vm-3:~$ sudo rabbitmqctl list_queues
Timeout: 60.0 seconds ...
Listing queues for vhost / ...
name      messages
voto_cancelacion      0
```