

Ya teniendo instalado docker y docker compose procedemos a correr el comando **docker -v** podemos identificar la version de docker que tenemos en nuestra maquina.

```
alvaro@VANDAL:~$ docker -v
Docker version 19.03.4, build 9013bf583a
alvaro@VANDAL:~$
```

Ahora procedemos a crear el contenedor de postgres que será el encargado de administrar nuestra base de datos

```
version: '3'
services:
  postgres:
    image: postgres:9.6
    container_name: postgres
    environment:
      - POSTGRES_DB=rolloutmaster
      - POSTGRES_USER=root
      - POSTGRES_PASSWORD=root
    ports:
      - "5432:5432"
    volumes:
      - /home/postgres_data:/var/lib/postgresql/data/
```

Levantamos nuestro contenedor con la base de datos usando el comando **sudo docker-compose up**

```

alvaro@VANDAL:~/Desktop/oscles$ sudo docker-compose up
Creating network "oscles_default" with the default driver
Pulling postgres (postgres:9.6)...
9.6: Pulling from library/postgres
80369df48736: Pull complete
b18dd0a6efec: Pull complete
5c20c5b8227d: Pull complete
c5a7f905c8ec: Pull complete
5a3f55930dd8: Pull complete
ffc097878b09: Pull complete
3106d02490d4: Pull complete
88d1fc513b8f: Pull complete
77bcb9a8971f: Pull complete
a1d54d716706: Pull complete
50a08c3d34b8: Pull complete
cda8b5ca36c7: Pull complete
643276ec28f6: Pull complete
f1653f203925: Pull complete
Digest: sha256:3d2856d01428052f1d4f6efa6ef78b15149c69e9574b9dae1488e3bed19abc4b
Status: Downloaded newer image for postgres:9.6
Creating postgres ... done
Attaching to postgres
postgres | The files belonging to this database system will be owned by user "postgres".
postgres | This user must also own the server process.
postgres |
postgres | The database cluster will be initialized with locale "en_US.utf8".
postgres | The default database encoding has accordingly been set to "UTF8".
postgres | The default text search configuration will be set to "english".
postgres |
postgres | Data page checksums are disabled.
postgres |
postgres | fixing permissions on existing directory /var/lib/postgresql/data ... ok
postgres | creating subdirectories ... ok
postgres | selecting default max_connections ... 100
postgres | selecting default shared_buffers ... 128MB
postgres | selecting default timezone ... Etc/UTC
postgres | selecting dynamic shared memory implementation ... posix
postgres | creating configuration files ... ok

```

En los logs de docker podemos identificar que el contenedor de postgres esta listo

```

alvaro@VANDAL:~/Desktop/oscles$ sudo docker-compose up -d
Starting postgres ... done

```

```

alvaro@VANDAL:~/Desktop/oscles$ sudo docker-compose ps

```

Name	Command	State	Ports
postgres	docker-entrypoint.sh postgres	Up	0.0.0.0:5432->5432/tcp

```

alvaro@VANDAL:~/Desktop/oscles$

```

```

postgres | selecting default shared_buffers ... 128MB
postgres | selecting default timezone ... Etc/UTC
postgres | selecting dynamic shared memory implementation ... posix
postgres | creating configuration files ... ok
postgres | running bootstrap script ... ok
postgres | performing post-bootstrap initialization ... ok
postgres | syncing data to disk ...
postgres | WARNING: enabling "trust" authentication for local connections
postgres | You can change this by editing pg_hba.conf or using the option -A, or
postgres | --auth-local and --auth-host, the next time you run initdb.
postgres | ok
postgres |
postgres | Success. You can now start the database server using:
postgres |
postgres |     pg_ctl -D /var/lib/postgresql/data -l logfile start
postgres |
postgres | waiting for server to start...LOG:  database system was shut down at 2019-11-05 23:29:42 UTC
postgres | LOG:  MultiXact member wraparound protections are now enabled
postgres | LOG:  database system is ready to accept connections
postgres | LOG:  autovacuum launcher started
postgres | done
postgres | server started
postgres | CREATE DATABASE
postgres |
postgres | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
postgres |
postgres | LOG:  received fast shutdown request
postgres | LOG:  aborting any active transactions
postgres | waiting for server to shut down...LOG:  autovacuum launcher shutting down
postgres | LOG:  shutting down
postgres | LOG:  database system is shut down
postgres | done
postgres | server stopped
postgres |
postgres | PostgreSQL init process complete; ready for start up.
postgres |
postgres | LOG:  database system was shut down at 2019-11-05 23:29:43 UTC
postgres | LOG:  MultiXact member wraparound protections are now enabled
postgres | LOG:  autovacuum launcher started
postgres | LOG:  database system is ready to accept connections

```

Configuramos nuestro repositorio en github para que todo marche bien cuando hagamos el contenedor de la aplicación web

AQUI VA LA IMAGEN DEL REPOSITORIO DE GITHUB

Ahora configuramos el docker file para django

```

## CONFIGURE LOCALES
RUN apt-get update && apt-get install -y locales && \
    sed -i 's%^# en_US.UTF-8 UTF-8$en_US.UTF-8 UTF-8%' /etc/locale.gen && \
    locale-gen
ENV LANG en_US.UTF-8

## UPDATE REPOSITORIES, INSTALL PIP AND BUILD DEPENDENCIES
RUN apt-get update && \
    apt-get install -y \
        zlib1g-dev \
        libpq-dev \
        libpcr3 \
        libpcr3-dev \
        git \
        wget \
        curl \
        libjpeg62-turbo-dev \
        python-dev \
        python3-dev \
        default-libmysqlclient-dev \
        python-mysqldb \
        vim

RUN cd /tmp/ && git clone https://github.com/oscles/API-sales.git && \
    pip install -r API-sales/requirements.txt \
    uwsgi

##COPY ENTRYPOINT SCRIPT
COPY django-entrypoint.sh /docker-entrypoint.sh

## CREATE A SPACE TO MOUNT DATA AND MOVE TO IT
RUN mkdir /app
WORKDIR /app

## DO NOT RUN CELERY / GUNICORN AS ROOT
RUN groupadd -g 1000 -r django && useradd -r -u 1000 -g django -s /bin/bash -m -d /home/django django

```

Nuestro archivo docker-compose se vería así:

```

version: '3'
services:
  postgres:
    image: postgres:9.6
    container_name: postgres
    environment:
      - POSTGRES_DB=rolloutmaster
      - POSTGRES_USER=root
      - POSTGRES_PASSWORD=root
    ports:
      - "5432:5432"
    volumes:
      - /home/postgres_data:/var/lib/postgresql/data/

  django:
    build:
      context: .
      dockerfile: Dockerfile.django
    container_name: django_app
    volumes:
      - /home/alvaro/Desktop/oscles/API-sales:/app/
    ports:
      - "8000:8000"
    entrypoint: ['/docker-entrypoint.sh']
    restart: on-failure
    depends_on:
      - postgres

```

Configuramos nuestro archivo docker-entrypoint.sh para que ejecute todos los comandos que tiene cuando se este construyendo el contenedor

```
#!/usr/bin/env bash

set -e

APPDIR="/app/"
APPNAME="API-sales"

cd $APPDIR

if [ "$1" = 'uwsgi' ] || [ "${#}" -eq 0 ]; then
    echo -n "[INFO] Running 'python manage.py collectstatic'..."
    python manage.py collectstatic --no-input >/dev/null
    echo "[OK]"
    ## custom command
    ##python manage.py makemigrations --merge
    echo -n "[INFO] Running 'python manage.py migrate'..."
    python manage.py migrate
    ## start uwsgi
    uwsgi --master --http 0.0.0.0:8000 --chdir /app/ --wsgi-file /app/comprame/wsgi.py \
        --static-map /static=/app/static/ --static-map /media=/app/media/ \
        --uid django --gid django --processes 5
else
    exec "$@"
fi
```

Volvemos a levantar nuestros contenedores para ver si todo esta funcionando bien con todas las dependencias que especificamos en los docker files y docker compose

```
alvaro@VANDAL:~/Desktop/oscles$ sudo docker-compose up
Building django
Step 1/10 : FROM python:3.6-stretch
3.6-stretch: Pulling from library/python
9a0b0ce99936: Pull complete
db3b6004c61a: Pull complete
f8f075920295: Pull complete
6ef14aff1139: Pull complete
0bbd8b48260f: Pull complete
b35072bbc91e: Pull complete
2a114127ad2d: Pull complete
bb188ce9189c: Pull complete
0c23e11bc19b: Pull complete
Digest: sha256:2f9a31ab290cc6bd80a0ce98d8b8f41777337669c6acb33616640ecfb7af8b6f
Status: Downloaded newer image for python:3.6-stretch
--> 61ef22d3ce5f
```

Tuvimos este error que indica que el contenedor de la aplicacion web ya existe necesitamos reconstruirlo para que todo funcione bien.

```
--> Running in ecd7bf77852
Removing intermediate container ecd7bf77852
--> 589b1deedf8
Step 9/10 : RUN groupadd -g 1000 -r django && useradd -r -u 1000 -g django -s /bin/bash -m -d /home/django django
--> Running in 5d364fb57da3
Removing intermediate container 5d364fb57da3
--> 331242d114d4
Step 10/10 : USER django
--> Running in 6138e726f4ee
Removing intermediate container 6138e726f4ee
--> 0ab17e7a6deb
Successfully built 0ab17e7a6deb
Successfully tagged oscles/django:latest
WARNING: Image for service django was built because it did not already exist. To rebuild this image you must use 'docker-compose build' or 'docker-compose up --build'.
postgres is up-to-date
Creating django_app ... error
ERROR: for django_app Cannot start service django: OCI runtime create failed: container_linux.go:346: starting container process caused 'exec: \"/docker-entrypoint.sh\": permission denied': unknown
ERROR: for django Cannot start service django: OCI runtime create failed: container_linux.go:346: starting container process caused 'exec: \"/docker-entrypoint.sh\": permission denied': unknown
ERROR: Encountered errors while bringing up the project.
alvaro@VANDAL:~/Desktop/oscles$
```

Ahora configuramos la base de datos en nuestro repositorio de la aplicación web en github


```
# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'rollout',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST': 'postgres',
        'PORT': 5432,
        'CHARSET': 'UTF-8'
    }
}
```

Ya con esa configuración realizada los contenedores deberían levantarse bien, ahora entramos al bash del contenedor de la aplicación web para ver si todo está corriendo bien haciendo el levantamiento del servicio de manera manual y podemos observar que está corriendo nuestra aplicación web con normalidad.

```
(venv) alvaro@VANDAL:~/Desktop/oscles$ sudo docker exec -it django_app bash
django@ebedcf5fbdde:/app$ ls
comprame db.sqlite3 manage.py README.md requirements.txt static
django@ebedcf5fbdde:/app$ python manage.py runserver 8000
Performing system checks...

System check identified no issues (0 silenced).
November 06, 2019 - 00:19:38
Django version 1.11.3, using settings 'comprame.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Error: That port is already in use.
django@ebedcf5fbdde:/app$ python manage.py runserver 7777
Performing system checks...

System check identified no issues (0 silenced).
November 06, 2019 - 00:19:46
Django version 1.11.3, using settings 'comprame.settings'
Starting development server at http://127.0.0.1:7777/
Quit the server with CONTROL-C.
```

57:01

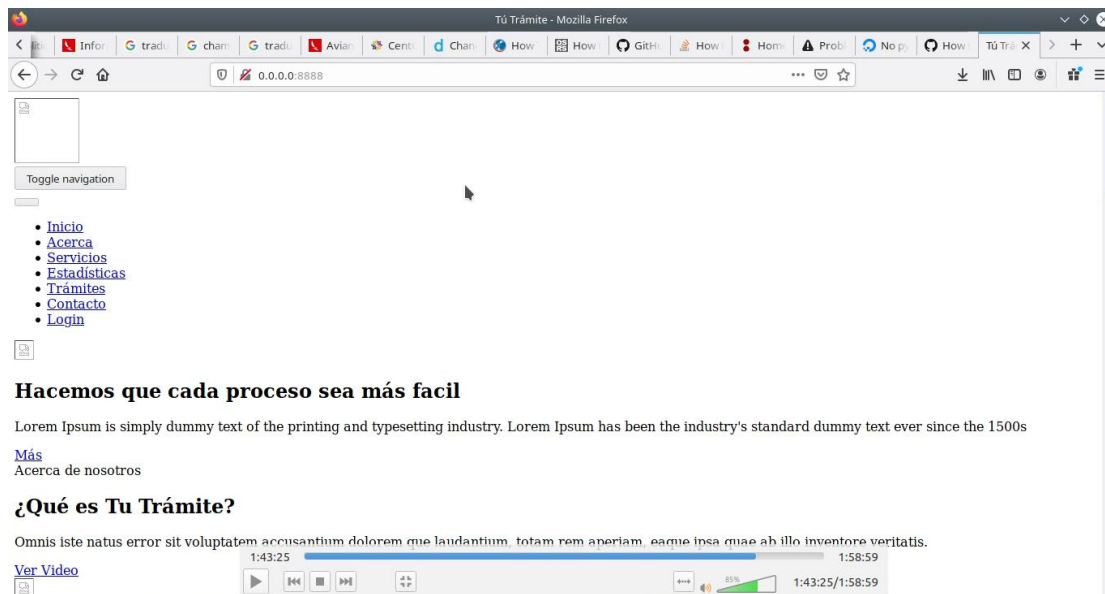
Ahora nuestros contenedores están corriendo

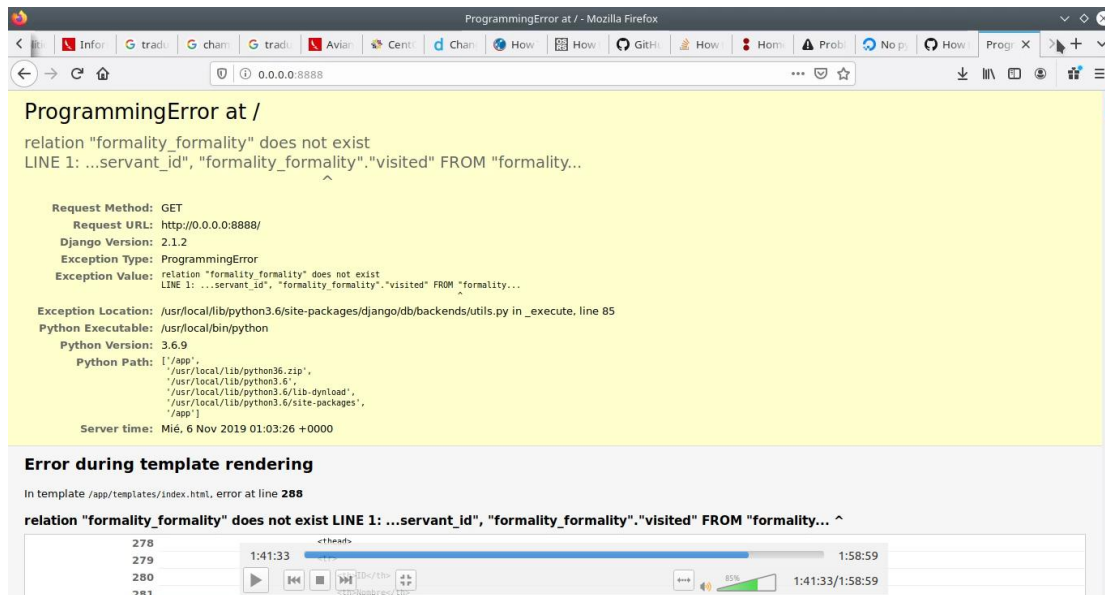
```

alvaro@VANDAL:~/Desktop/oscles$ sudo docker-compose up
Starting postgres ... done
Recreating django_app ... done
Attaching to postgres, django_app
postgres | LOG: database system was shut down at 2019-11-06 00:53:10 UTC
postgres | LOG: MultiXact member wraparound protections are now enabled
postgres | LOG: database system is ready to accept connections
postgres | LOG: autovacuum launcher started
django_app | [INFO] Running 'python manage.py collectstatic'...[OK]
django_app | [INFO] Running 'python manage.py migrate'...SystemCheckError: System check identified some issues:
django_app | 
django_app | ERRORS:
django_app | ?: (staticfiles.E002) The STATICFILES_DIRS setting should not contain the STATIC_ROOT setting.
django_app | exited with code 1
django_app | [INFO] Running 'python manage.py collectstatic'...[OK]
django_app | SystemCheckError: System check identified some issues:
django_app | 
django_app | ERRORS:
django_app | ?: (staticfiles.E002) The STATICFILES_DIRS setting should not contain the STATIC_ROOT setting.
django_app | [INFO] Running 'python manage.py migrate'...[INFO] Running 'python manage.py collectstatic'...[OK]
django_app | SystemCheckError: System check identified some issues:
django_app | 
django_app | ERRORS:
django_app | ?: (staticfiles.E002) The STATICFILES_DIRS setting should not contain the STATIC_ROOT setting.
django_app | [INFO] Running 'python manage.py migrate'...[INFO] Running 'python manage.py collectstatic'...[OK]
django_app | SystemCheckError: System check identified some issues:
django_app | 
django_app | ERRORS:
django_app | ?: (staticfiles.E002) The STATICFILES_DIRS setting should not contain the STATIC_ROOT setting.
^CGracefully stopping... (press Ctrl+C again to force)
Stopping django_app ... done
Stopping postgres ... done

```

Pero tenemos el problema de que no esta cargando los estilos de nuestra aplicación web, y cuando estramos nos dice que no se han cargado las migraciones de la base de datos mostrando el siguiente error:





Solucionamos el problema de las migraciones

```
(venv) → apps git:(master) python ../manage.py makemigrations formality
Migrations for 'formality':
  formality/migrations/0001_initial.py
  - Create model Formality
(venv) → apps git:(master) python ../manage.py makemigrations entity
Migrations for 'entity':
  entity/migrations/0001_initial.py
  - Create model Entity
(venv) → apps git:(master) python ../manage.py makemigrations civil_servant
Migrations for 'civil_servant':
  civil_servant/migrations/0001_initial.py
  - Create model CivilServant
(venv) → apps git:(master) python ../manage.py makemigrations attachment
Migrations for 'attachment':
  attachment/migrations/0001_initial.py
  - Create model AttachmentFormality
(venv) → apps git:(master)
```

Colocamos los comandos que ejecutamos manualmente en nuestro docker-entrypoint.sh para que nuestro contenedor corra bien


```
#!/usr/bin/env bash

set -e

APPDIR="/app/"
APPNAME="tramite"

cd $APPDIR

if [ "$1" = 'uwsgi' ] || [ "${#}" -eq 0 ]; then
    echo -n "[INFO] Running 'python manage.py collectstatic'..."
    python manage.py collectstatic --no-input >/dev/null
    echo "[OK]"
    ## custom command
    ##python manage.py makemigrations --merge
    echo -n "[INFO] Running 'python manage.py migrate'..."
    python manage.py migrate
    python manage.py makemigrations formality
    python manage.py makemigrations entity
    python manage.py makemigrations civil_servant
    python manage.py makemigrations attachment
    echo "runserver working"
    ## start uwsgi
    uwsgi --master --http 0.0.0.0:8000 --chdir /app/ --wsgi-file /app/tramite/wsgi.py \
        --static-map /static=/app/static/ --static-map /media=/app/media/ \
        --uid django --gid django --processes 5
else
    exec "$@"
fi
```

Ahora levantamos nuestros contenedores nuevamente con nuestra aplicación corriendo perfectamente

```

alvaro@VANDAL:~/Desktop/oscles$ sudo docker-compose build --no-cache
postgres uses an image, skipping
Building django
Step 1/10 : FROM python:3.6-stretch
--> 61ef22d3ce5f
Step 2/10 : RUN apt-get update && apt-get install -y locales && sed -i 's%^# en_US.UTF
-8 UTF-8$%en_US.UTF-8 UTF-8%' /etc/locale.gen && locale-gen
--> Running in 07d08a26dd65
Get:1 http://security.debian.org/debian-security stretch/updates InRelease [94.3 kB]
Ign:2 http://deb.debian.org/debian stretch InRelease
Get:3 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [500 kB]
Get:5 http://deb.debian.org/debian stretch Release [118 kB]
Get:6 http://deb.debian.org/debian stretch-updates/main amd64 Packages [27.4 kB]
Get:7 http://deb.debian.org/debian stretch Release.gpg [2365 B]
Get:8 http://deb.debian.org/debian stretch/main amd64 Packages [7086 kB]
Fetched 7919 kB in 6s (1295 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  libc-l10n
The following NEW packages will be installed:
  libc-l10n locales
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 4108 kB of archives.
After this operation, 13.8 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 libc-l10n all 2.24-11+deb9u4 [820 kB]
Get:2 http://deb.debian.org/debian stretch/main amd64 locales all 2.24-11+deb9u4 [3289 kB]
debconf: delaying package configuration, since apt-utils is not installed
Fetched 4108 kB in 1s (2803 kB/s)
Selecting previously unselected package libc-l10n.
(Reading database ... 30537 files and directories currently installed.)
Preparing to unpack .../libc-l10n_2.24-11+deb9u4_all.deb ...
Unpacking libc-l10n (2.24-11+deb9u4) ...
Selecting previously unselected package locales.
Preparing to unpack .../locales_2.24-11+deb9u4_all.deb ...

```

