

Delivery #1: Discovery & Reverse Engineering

TASK #1 Onboarding Log:

Información de versiones y dependencias estrictamente necesarias para levantar exitosamente nuestro proyecto:

- Python v3.5.2 / v3.6.15.
- Django v1.10.5
- pyenv v3.5.2
- pyenv-virtualenv v3.5.2

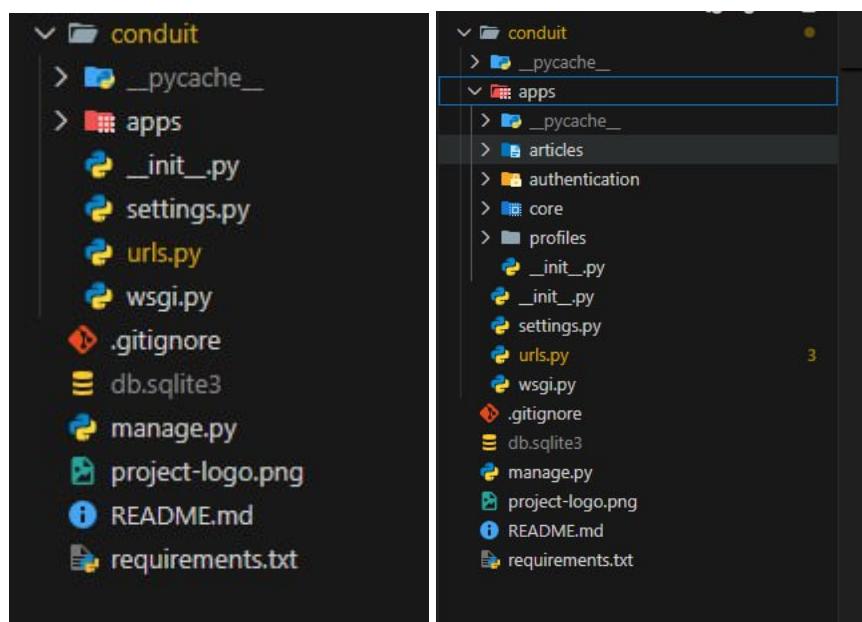
Se hizo el fork del proyecto: <https://github.com/Daniel-1A/django-realworld-example-app> a los integrantes del grupo se les envió una invitación al repositorio para que sean colaboradores y todos puedan trabajar.

Se clonó el repositorio en nuestra máquina local, al ingresar al archivo [README.md](#) encontramos una serie de pasos de para la instalación de dependencias que iban a funcionar para levantar el proyectos.

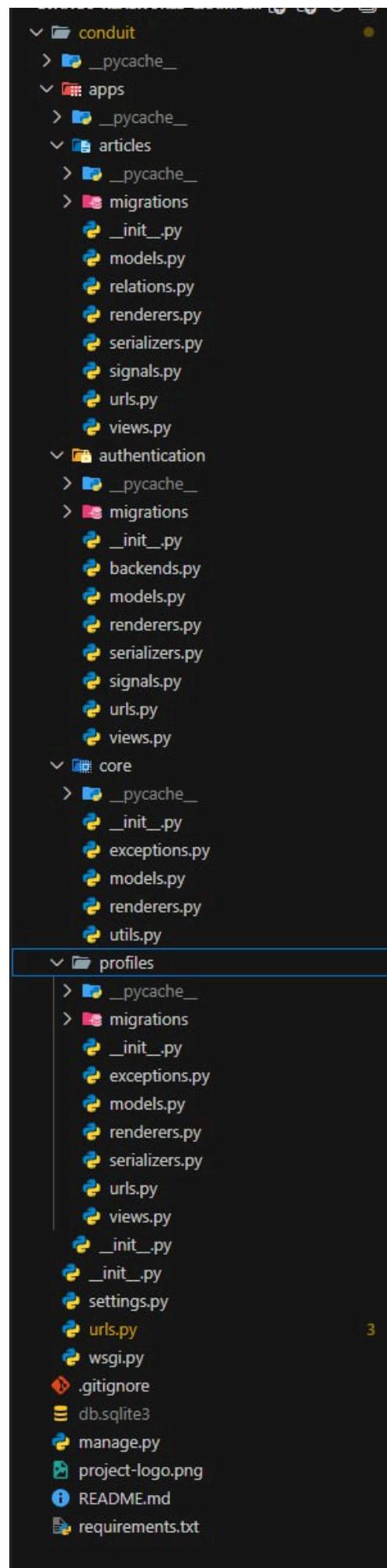
Problema #1:

Encontramos unas carpetas que no existen el proyecto, específicamente las carpetas:

- conduit-django
- productionready-django-api



Carlos Daniel Martínez García - 19001064
Douglas Daniel Pérez Hernández - 22003865



Carlos Daniel Martínez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Solución:

Para esto, nos apoyamos con un LLM (Copilot) para poder ir descubriendo la estructura del proyecto y como poder levantararlo.

Problema #2:

Intentamos levantar el proyecto con versiones más recientes de python y django pero nos topamos con errores de librerías que solo funcionaban con las versiones del README entonces indagando más sobre el problema descubrimos que nuestra versión (actual) de python no es compatible con la versión de Django que el proyecto utiliza, por lo que procedimos a instalar las versiones que se nos indicaba en el readme, esto es un desafío ya que la versión de python es antigua y ya no está disponible.

```
> PS C:\Users\daniel\Downloads\GitHub\django-realworld-example-app> pip install --requirements.txt
Collecting Django==1.10.5 (from --requirements.txt (line 1))
  Using cached Django-1.10.5-py2.py3-none-any.whl
    metadata (1.3 kB)
  Collected django-cors-middleware==1.3.1 (from --requirements.txt (line 2))
    Using cached django_cors_middleware-1.3.1-py3-none-any.whl
    Collected django-extensions==1.7.1 (from --requirements.txt (line 3))
    Using cached django_extensions-1.7.1-py2.py3-none-any.whl.metadata (1.0 kB)
  Collected django-rest-framework==3.4.4 (from --requirements.txt (line 4))
    Using cached django_rest_framework-3.4.4-py2.py3-none-any.whl
    metadata (8.9 kB)
  Collected PyJWT==1.4.2 (from --requirements.txt (line 5))
    Using cached PyJWT-1.4.2-py2.py3-none-any.whl.metadata (2.7 kB)
  Collected six==1.10.0 (from --requirements.txt (line 6))
    Using cached six-1.10.0-py2.py3-none-any.whl (1.3 kB)
  Collected Django==1.10.5-py2.py3-none-any.whl (6.8 kB)
  Using cached django_extensions-1.7.1-py2.py3-none-any.whl (200 kB)
Using cached django-rest-framework-3.4.4-py2.py3-none-any.whl (70 kB)
Using cached PyJWT-1.4.2-py2.py3-none-any.whl (15 kB)
Using cached six-1.10.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: six, PyJWT, django-rest-framework, django-cors-middleware, Django, django-extensions
Successfully installed Django-1.10.5 PyJWT-1.4.2 django-cors-middleware-1.3.1 django-rest-framework-3.4.4 six-1.10.0
[notice] A new release of pip is available: 25.0.1 -> 26.0 [notice] To update, run: python -m pip install --upgrade pip
(venv) PS C:\Users\daniel\Downloads\GitHub\django-realworld-example-app> python -m django --version
Traceback (most recent call last):
  File "<frozen r奴py>", line 198, in __run_code
    exec(code, self._global_ns, self._local_ns)
  File "<frozen r奴py>", line 108, in __file__
    from django.core import management
  File "C:\Users\daniel\Downloads\GitHub\django-realworld-example-app\venv\lib\site-packages\django\core\management\_init_.py", line 1, in <module>
    from django.apps import apps
  File "C:\Users\daniel\Downloads\GitHub\django-realworld-example-app\venv\lib\site-packages\django\apps\__init__.py", line 1, in <module>
    from django.core.exceptions import AppConfigReady, ImproperlyConfigured
  File "C:\Users\daniel\Downloads\GitHub\django-realworld-example-app\venv\lib\site-packages\django\core\exceptions.py", line 5, in <module>
    from django.utils.encoding import force_text
  File "C:\Users\daniel\Downloads\GitHub\django-realworld-example-app\venv\lib\site-packages\django\utils\encoding.py", line 11, in <module>
    from django.utils.six.moves.urllib.parse import quote, unquote
ModuleNotFoundError: No module named 'django.utils.six.moves'
```

Python 3.5.10

Warning: Python 3.5.10 reached end-of-life on 2020-09-30. It is no longer supported and does not receive security updates. We recommend upgrading to the latest Python release.

Release date: Sept. 5, 2020

Python 3.5.10

Python 3.5 has reached end-of-life. Python 3.5.10 is the final release of 3.5.

Python 3.5.10 was released on September 5th, 2020.

Python 3.5.10 is the final release in the Python 3.5 series. As of this release, the 3.5 branch has been retired, no further changes to 3.5 will be accepted, and no new releases will be made. This is standard Python policy; Python releases get five years of support and are then retired.

If you're still using Python 3.5, you should consider upgrading to the [current version](#). Newer versions of Python have many new features, performance improvements, and bug fixes, which should all serve to enhance your Python programming experience.

We in the Python core development community thank you for your interest in 3.5, and we wish you all the best!

Major new features of the 3.5 series, compared to 3.4

Solución:

Utilizando una imagen de Docker logramos hacer que la versión de Python fuera compatible con la versión de Django y logramos correr el proyecto, para ello fue necesario realizar los siguientes pasos:

1. Descargar Docker
2. En la carpeta raíz del proyecto ejecutamos el siguiente comando:
 - **docker run -it -v \${PWD}:/app -w /app python:3.6 bash**

```
Unable to find image 'python:3.6' locally
3.6: Pulling from library/python
0e29546d541c: Pull complete
404f02044bac: Pull complete
6494e4811622: Pull complete
cb5b7ae36172: Pull complete
c4f42be2be53: Pull complete
5e3b1213efc5: Pull complete
9fddfdc56334: Pull complete
9b829c73b52b: Pull complete
6f9f74896dfa: Pull complete
Digest: sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
Status: Downloaded newer image for python:3.6
docker: Error response from daemon: create ${PWD}: "${PWD}" includes invalid characters for a local volume name, only "[a-zA-Z0-9][a-zA-Z0-9_.-]" are allowed. If you intended to pass a host directory, use absolute path
Run 'docker run --help' for more information
```

3. Luego de ejecutar el comando, se levantará una sesión “root”, donde tendremos que ejecutar el siguiente comando:
- **pip install -r requirements.txt**

```
root@29f3f5d61abf:/app# pip install -r requirements.txt
Collecting Django==1.10.5
  Downloading Django-1.10.5-py2.py3-none-any.whl (6.8 MB)
    ...
Collecting django-cors-middleware==1.3.1
  Downloading django-cors-middleware-1.3.1.tar.gz (9.0 kB)
Collecting django-extensions==1.7.1
  Downloading django_extensions-1.7.1-py2.py3-none-any.whl (200 kB)
Collecting djangorestframework==3.4.4
  Downloading djangorestframework-3.4.4-py2.py3-none-any.whl (706 kB)
    ...
Collecting PyJWT==1.4.2
  Downloading PyJWT-1.4.2-py2.py3-none-any.whl (15 kB)
Collecting six==1.10.0
  Downloading six-1.10.0-py2.py3-none-any.whl (10 kB)
Building wheels for collected packages: django-cors-middleware
  Building wheel for django-cors-middleware (setup.py) ... done
    Created wheel for django-cors-middleware: filename=django_cors_middleware-1.3.1-py3-none-any.whl size=8514 sha256=51b38487221e5a25a200cc7694e527b049616165f834aea00357895fe7a918df
  Stored in directory: /root/.cache/pip/wheels/8d/10/9d/b56074b461ac2502132cf453f42446811a382dfa5b9635c8f
Successfully built django-cors-middleware
Installing collected packages: six, PyJWT, djangorestframework, django-extensions, django-cors-middleware, Django
Successfully installed Django-1.10.5 PyJWT-1.4.2 django-cors-middleware-1.3.1 django-extensions-1.7.1 djangorestframework-3.4.4 six-1.10.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

4. Para corroborar que todo esté bien, necesitamos validar que las versiones son las que el proyecto requiere en este caso la que más importa es tener la versión de django tal y como lo dice el README para el python basta con que sea compatible para la versión de Django, estas son las versiones que estamos utilizando;
- **python -m django --version**

```
root@29f3f5d61abf:/app# python -m django --version
1.10.5
root@29f3f5d61abf:/app# python --version
Python 3.6.15
```

5. Hacemos las migraciones correspondientes:
- **python manage.py migrate**

```
root@29f3f5d61abf:/app# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, articles, auth, authentication, contenttypes, profiles, sessions
Running migrations:
  No migrations to apply.
```

Problema #3:

Una vez solucionado lo de las versiones y demás, procedimos a correr el proyecto, en este punto todavía no tenemos idea de que hace el código, pero intuimos que al tratarse de un proyecto con Python y Django probablemente lo que tengamos sea un backend, para ello ejecutamos el comando:

- **python manage.py runserver 0.0.0.0:8000**

Vimos esta información desplegada en consola:

```
root@29f3f5d61abf:/app# python manage.py runserver 0.0.0.0:8000
Performing system checks...

System check identified no issues (0 silenced).
February 03, 2026 - 23:48:01
Django version 1.10.5, using settings 'conduit.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Con esta imagen nos damos cuenta de que efectivamente es un backend, lo que ya sabemos es cómo comunicarnos con él para comenzar a descubrir qué es lo que hace.

Solución:

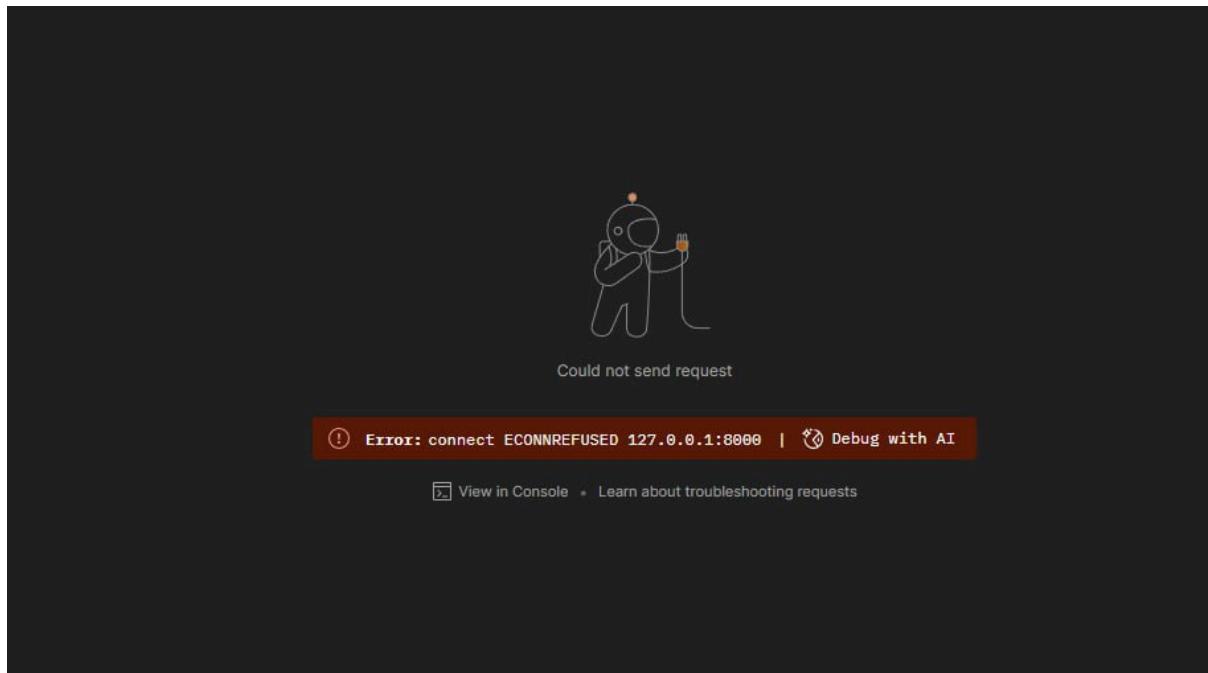
Utilizando el paquete django-extensions podemos obtener el listado de los endpoints que forman parte del proyecto tal y como se muestra en la siguiente imagen, por lo que ya sabemos como comunicarnos con este backend, solamente hace falta probarlo para ver si nos responde:

```
root@ef1fdc1441fc:/app# python manage.py show_urls
/admin/ django.contrib.admin.sites.index      admin:index
/admin/<app_label>/    django.contrib.admin.sites.app_index    admin:app_list
/admin/auth/group/    django.contrib.admin.options.changelist_view    admin:auth_group_changelist
/admin/auth/group/<var>/    django.views.generic.base.RedirectView
/admin/auth/group/<var>/change/  django.contrib.admin.options.change_view    admin:auth_group_change
/admin/auth/group/<var>/delete/  django.contrib.admin.options.delete_view    admin:auth_group_delete
/admin/auth/group/<var>/history/   django.contrib.admin.options.history_view    admin:auth_group_history
/admin/auth/group/add/  django.contrib.admin.options.add_view    admin:auth_group_add
/admin/jsi18n/    django.contrib.admin.sites.i18n_javascript    admin:jsi18n
/admin/login/     django.contrib.admin.sites.login    admin:login
/admin/logout/    django.contrib.admin.sites.logout    admin:logout
/admin/password_change/  django.contrib.admin.sites.password_change    admin:password_change
/admin/password_change/done/  django.contrib.admin.sites.password_change_done    admin:password_change_done
/admin/r/<content_type_id>/<object_id>/  django.contrib.contenttypes.views.shortcut    admin:view_on_site
/api/  rest_framework.routers.APIRoot  articles:api-root
/api/<format>/ rest_framework.routers.APIRoot  articles:api-root
/api/articles/ conduit.apps.articles.views.ArticleViewSet    articles:article-list
/api/articles.<format>/ conduit.apps.articles.views.ArticleViewSet    articles:article-list
/api/articles/<article_slug>/comments/  conduit.apps.articles.views.CommentsListCreateAPIView
/api/articles/<article_slug>/comments/<comment_pk>/  conduit.apps.articles.views.CommentsDestroyAPIView
/api/articles/<article_slug>/favorite/  conduit.apps.articles.views.ArticlesFavoriteAPIView
/api/articles/<slug>    conduit.apps.articles.views.ArticleViewSet    articles:article-detail
/api/articles/<slug>.<format>/  conduit.apps.articles.views.ArticleViewSet    articles:article-detail
/api/articles/feed/    conduit.apps.articles.views.ArticlesFeedAPIView
/api/profiles/<username>/    conduit.apps.profiles.views.ProfileRetrieveAPIView
/api/profiles/<username>/follow/  conduit.apps.profiles.views.ProfileFollowAPIView
/api/tags/    conduit.apps.articles.views.TagListAPIView
/api/user/    conduit.apps.authentication.views.UserRetrieveUpdateAPIView
/api/users/    conduit.apps.authentication.views.RegistrationAPIView
/api/users/login/    conduit.apps.authentication.views.LoginAPIView
root@ef1fdc1441fc:/app# python manage.py runserver 0.0.0.0:8000
Performing system checks...
```

Carlos Daniel Martínez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Problema #4:

Al tratar de realizar pruebas a través de postman, nos topamos con el siguiente problema de conexión, no logramos comunicarnos con ningún endpoint del proyecto.



Solución:

Este problema está ligado a que al momento de inicializar docker no definimos ningún puerto para el proyecto, es por esto mismo que teníamos el error de conexión, al establecer el puerto 8000 se soluciona el error de conexión y logramos ya tener el backend arriba.

A screenshot showing two side-by-side interfaces. On the left is the Postman application, displaying a GET request to "http://localhost:8000/api/tags". The response body shows a JSON object with an empty "tags" array. On the right is a web browser window titled "localhost8000/api/articles", showing the same JSON response: {"articles": [], "articlesCount": 0}. The browser also displays the status bar with "Impresión con formato estilístico" and a "Inicio" button.

Aquí algunos ejemplos de algunos endpoints que probamos tanto en postman como en nuestro navegador.

Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Comandos para levantar el proyecto:

1. docker run -it -p 8000:8000 -v \${PWD}:/app -w /app python:3.6 bash
2. pip install -r requirements.txt
3. python -m django --version
4. python manage.py migrate
5. python manage.py runserver 0.0.0.0:8000