

OPEN EDX

Oriana Pettinelli & Lara Noeli Cardozo

Open edX es una plataforma de aprendizaje en línea de código abierto, altamente escalable y personalizable, creada para desarrollar y entregar experiencias educativas a gran escala, desde Cursos Masivos Abiertos en Línea (MOOC) hasta programas de grado completos, utilizada por universidades, empresas y gobiernos. Permite crear cursos interactivos con foros, videos, evaluaciones y análisis en tiempo real, impulsando un ecosistema de aprendizaje centrado en el estudiante.

Sus dos features principales son:

LMS (Learning Management System):

La interfaz principal para estudiantes y administradores, permitiendo acceder a cursos, participar en foros, verificar progreso y gestionar inscripciones.

CMS: (Content Management System):

La herramienta para que los instructores creen y editen cursos, agreguen contenido multimedia, establezcan políticas de calificación y gestionen el equipo del curso.

La plataforma está escrita en Python y JavaScript y hace un uso extensivo del framework Django. En su nivel más alto, se compone de un monolito modular, algunas aplicaciones de implementación independiente y micro frontends basados en ReactJS.

El repositorio nos ofrece dos alternativas para instalar openEdx, la versión fuente "bare metal" o TUTOR, la distribución dockerizada, para este proyecto vamos a utilizar la primera.



INSTALACIÓN

Dependencias

OS:

Ubuntu 24.04

Lenguajes y herramientas:

Python 3.11

Node: See the .nvmrc file in this repository.

Servicios:

MySQL 8.0

Mongo 7.x

Memcached

Para instalar y correr la plataforma localmente, hay que seguir una serie de pasos, teniendo en cuenta que al ser una plataforma robusta requiere muchas dependencias.

En nuestro caso instalamos Edx en una computadora hp con Intel Core i3-7020U CPU @ 2.30GHz, 2 núcleos físicos, 8 gib de ram, 2gb de swap.

El sistema operativo es ubuntu, en la versión 24.04.3

En primer lugar queremos instalar el repositorio principal, Edx-platform.

Es clave destacar que el readme con instrucciones para instalarlo sin TUTOR, aclara que este es un método de instalación no mantenido activamente por la comunidad, y que es probable que muchos errores que uno se encuentra en el camino no están documentados, o la comunidad no sepa responder.

Menciona que sólo es recomendable para desarrolladores buscando una aventura o para administradores de sistemas experimentados que prefieren dejar en sus propias manos la complejidad de OpenEdx.

A continuación se listan los pasos que seguimos para realizar la instalación.

DESCARGAR PROYECTO

Git

```
sudo apt install git
```

Clonar el [repositorio](#)

DEPENDENCIAS

Mysql

```
sudo apt install mysql-server
```

Mongo

Chequear página oficial

Node y npm

```
sudo apt install nodeJs npm
```

Memcached

```
sudo apt install libmemcached-tools memcached
```

Pip (Asumimos qué ubuntu ya tiene instalado python)

```
python3 -m ensurepip --upgrade
```

Dependencias del sistema

```
sudo apt install -y \  
  python3-dev \  
  default-libmysqlclient-dev \  
  build-essential \  
  pkg-config \  
  libmemcached-dev \  
  zlib1g-dev \  
  libjpeg-dev \  
  libpng-dev \  
  libfreetype6-dev \  
  liblcms2-dev \  
  libwebp-dev \  
  tcl8.6-dev \  
  tk8.6-dev \  
  libharfbuzz-dev \  
  libfribidi-dev \  
  libxcb1-dev
```

Pyenv

```
curl https://pyenv.run | bash  
nano ~/.bashrc
```

Añade las líneas `export PATH="$HOME/.pyenv/bin:$PATH"` y `eval "$(pyenv init --path)"` al final del archivo, guardar y aplicar los cambios con `source ~/.bashrc` o reinicia la terminal.

Python

Necesitamos tener la versión 3.11 de python según indicaciones del proyecto, para esto usamos pyenv.

En una terminal localizada en la carpeta root del proyecto

```
pyenv install 3.11  
pyenv local 3.11
```

Chequeamos qué la versión sea la correcta:

```
python --version
```

SETUP

Virtual environment

Para este proyecto necesitamos crear un entorno virtual para instalar las dependencias. Lo creamos de la siguiente manera

```
python3 -m venv venv
```

Activamos el environment

```
source venv/bin/activate
```

Actualizamos paquetes para que vayan con la versión de python de este env

```
pip install --upgrade pip setuptools wheel
```

Instalar requirements

```
pip install -r requirements/edx/base.txt  
pip install -r requirements/edx/development.txt  
pip install -e .
```

```
npm clean-install --dev
```

Bases de datos

MONGO:

Edx necesita un usuario para conectarse a mongo

```
mongosh
```

```
use edxapp  
db.createUser({  
  user: "edxapp",  
  pwd: "password",  
  roles: [  
    { role: "readWrite", db: "edxapp" }  
  ]  
})
```

```
]
})
exit
```

Si desea cambiar la contraseña o el usuario debe asegurarse que estas coincidan con los valores en las configuraciones.

MYSQL

```
sudo systemctl start mysql
```

```
sudo mysql -u root
```

```
CREATE DATABASE edxapp
  CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;
```

```
CREATE DATABASE edxapp_csmh
  CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;
```

```
CREATE USER 'edxapp'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON edxapp.* TO 'edxapp'@'localhost';
GRANT ALL PRIVILEGES ON edxapp_csmh.* TO 'edxapp'@'localhost';
FLUSH PRIVILEGES;
```

```
EXIT;
```

Variables de entorno

Tendremos que actualizar los archivos `lms/envs/minimal` y `cms/envs/minimal`.

En estos archivos tendremos configuraciones adicionales para el proyecto. Siempre que nos referimos a configuraciones, hablamos de estos archivos.

Las configuraciones que usamos para la instalación se encuentran al final del documento.

Luego seteamos las variables de entorno de configuración para `lms` y `cms`, de la siguiente manera:

```
export LMS_CFG=$(path hasta el proyecto)/edx-platform/lms/envs/minimal.yml
export CMS_CFG=$(path hasta el proyecto)/edx-platform/cms/envs/minimal.yml
```

O podemos agregar estas mismas variables a un archivo `.env` en el root del proyecto

Migraciones

Necesitaremos correr las migraciones. (Siempre con el environment activado)

```
./manage.py lms migrate --skip-checks  
./manage.py lms migrate --database=student_module_history --skip-checks  
./manage.py cms migrate --skip-checks
```

Posible problema

Al instalar openedx y correr las migraciones, hay algunas apps de django que parecen no cargarse en el orden apropiado y generar un error de runtime. Por este motivo usamos *--skip-checks*.

Assets

```
npm run build-dev
```

Posible problema

Chequear dónde se generan los archivos estáticos, y donde espera Django encontrarlos. Hay una variable llamada `STATIC_ROOT` que es dónde deberán estar localizados los archivos generados.

El archivo que buscará es `webpack-stats.json`. Se puede mover este archivo a la ubicación esperada, o cambiar la configuración al directorio deseado.

Setup de SSO (Single sign on)

CMS SSO permite que los usuarios inicien sesión en Studio (CMS) usando sus credenciales del LMS. Esto se hace mediante autenticación OAuth2 donde:

- LMS actúa como el proveedor OAuth2 (servidor de autorización)
- CMS actúa como el cliente OAuth2 (aplicación que solicita acceso)

La aplicación necesita estar asociada a un usuario.

```
./manage.py lms manage_user studio_worker example@example.com --unusable-password  
--skip-checks
```

El flag `--unusable-password` significa que este usuario no puede iniciar sesión directamente (solo se usa para OAuth)

Ahora creamos la aplicación OAuth

```
./manage.py lms create_dot_application studio-sso-key studio_worker \  
--grant-type authorization-code \  
--skip-authorization \  
--skip-authorization
```

```
--redirect-uris 'http://127.0.0.1:18010/complete/edx-oauth2/' \  
--scopes user_id \  
--client-id 'studio-sso-key' \  
--client-secret 'studio-sso-secret' \  
--skip-checks
```

Iniciar la plataforma

Verificar qué Mysql, mongo y memcached están corriendo.

LMS:

```
./manage.py lms runserver 18000
```

CMS:

```
./manage.py cms runserver 18010
```

Crear super user

```
./manage.py lms shell -c "from django.contrib.auth.models import User; u =  
User.objects.create_user('admin', 'admin@example.com', 'admin123'); u.is_superuser =  
True; u.is_staff = True; u.save(); print('Superuser created: admin/admin123')"
```

Nos puede servir para acceder a toda la plataforma ya que no tiene limite de permisos. Igual podremos registrar un usuario a través del formulario de registro en la plataforma.

Micro frontends

OpenEdx tiene un gran listado de micro frontends para acceder a distintas partes de la plataforma.

En este caso vamos a instalar el requerido para crear cursos, [course authoring](#).

Tendremos que:

Clonar el repositorio

(Fuera del otro repositorio de edx)

```
git clone https://github.com/openedx/frontend-app-course-authoring.git
```

Instalar dependencias

```
npm install
```

Copiar variables de entorno:

En un archivo .env agregar las settings para nuestro proyecto.

LMS_BASE_URL=<http://127.0.0.1:18000>

STUDIO_BASE_URL=<http://127.0.0.1:18010>

BASE_URL=<http://127.0.0.1:2001>

PORT=2001

Levantar el proyecto

npm start

ALGUNAS CONFIGURACIONES EXTRA

Edx nos permite cambiar ciertos atributos de la plataforma a través de su archivo de configuraciones, para adaptarla a las necesidades.

Por ejemplo, agregando la variable PLATFORM_NAME podemos setear el nombre de la plataforma.

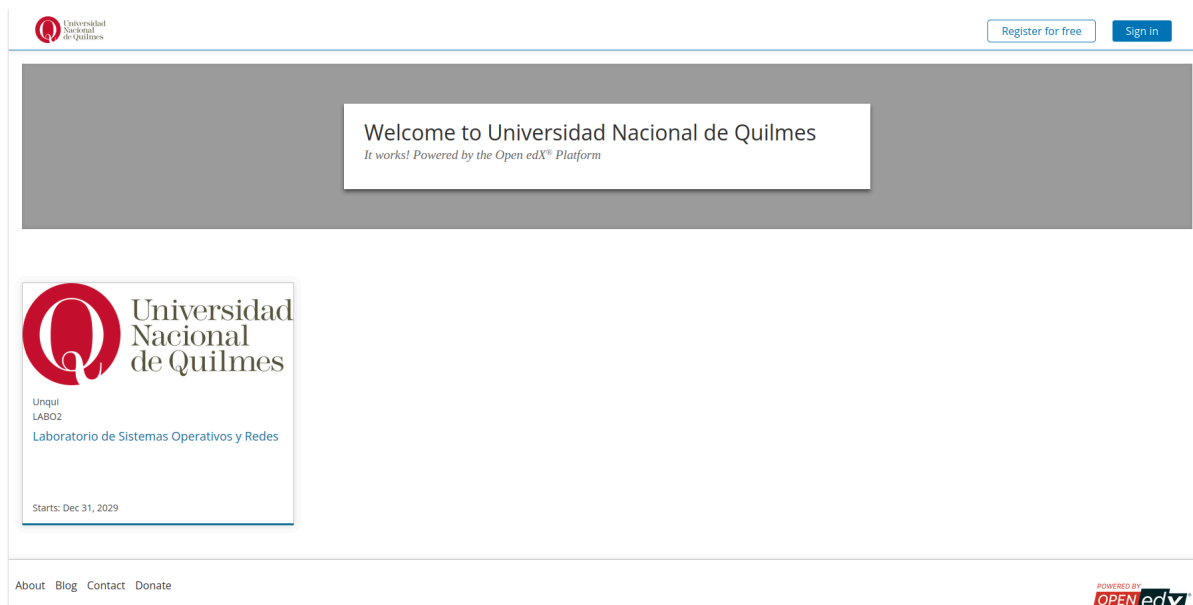
También si en los assets estáticos agregamos una imagen con el nombre "logo", tanto en lms como en cms, está se va a usar como logo de la plataforma.

El proyecto tiene un archivo llamado mock.yml que sirve de referencia para todas las posibles configuraciones que se pueden incluir en los archivos minimal.yml, o el archivo de configuración elegido.

COMO USAR

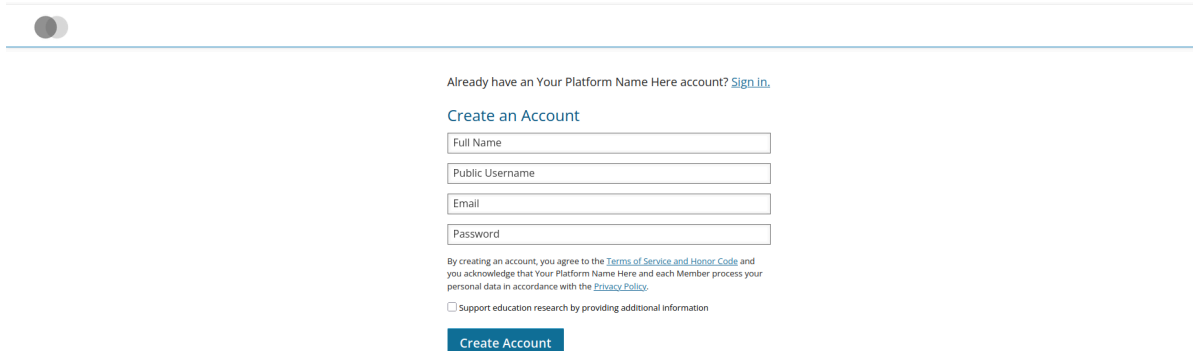
Entramos a <http://127.0.0.1:18000>

Veremos la home page, dónde se listan cursos creados. Si es la primera vez que levantas la plataforma, no habrá cursos creados.



Registrar un usuario

El botón de registrarse te redirecciona a un formulario dónde se puede crear una cuenta
Luego se puede utilizar está cuenta para iniciar sesión





First time here? [Create an Account.](#)

Sign In

Email

The email address you used to register with Your Platform Name Here

Password

*Need help signing in?

Sign in with your company or school

Sign In

DASHBOARD



Courses

Discover New

Help



LaraOrlana

My Courses

You are not enrolled in any courses yet.

Explore Courses

[About](#) [Blog](#) [Contact](#) [Donate](#)




© Your Platform Name Here. All rights reserved except where noted. edX, Open edX and their respective logos are registered trademarks of edX Inc.
[Terms of Service & Honor Code](#) - [Privacy Policy](#) - [Take free online courses at edX.org](#)



EXPLORAR CURSOS

Si presionamos en explorar cursos, podemos ver los cursos disponibles para que el usuario se anote

Universidad Nacional de Quilmes Discover New Help LaraOriana



Universidad Nacional de Quilmes

Unqui LABO2
Laboratorio de Sistemas Operativos y Redes

Starts: Dec 31, 2029

About Blog Contact Donate

POWERED BY OPEN edX

© Universidad Nacional de Quilmes. All rights reserved except where noted. edX, Open edX and their respective logos are registered trademarks of edX Inc.
Terms of Service & Honor Code - Privacy Policy - Take free online courses at edX.org

DETALLE DEL CURSO

En el detalle del curso podemos ver información relacionada, y un botón para anotarnos


Universidad Nacional de Quilmes Unqui LABO2 Laboratorio de Sistemas Operativos y Redes Discover New Help LaraOriana

Unqui

Laboratorio de Sistemas Operativos y Redes

Este es un curso de la materia de Laboratorio de sistemas operativos y redes de la universidad nacional de quilmes

[Enroll Now](#)



Universidad Nacional de Quilmes

[Twitter](#) [Facebook](#) [Email](#)

| | |
|---------------|--------------|
| Course Number | LABO2 |
| Price | Free |

About Blog Contact Donate

POWERED BY OPEN edX

© Universidad Nacional de Quilmes. All rights reserved except where noted. edX, Open edX and their respective logos are registered trademarks of edX Inc.
Terms of Service & Honor Code - Privacy Policy - Take free online courses at edX.org

Si presionamos enroll now, ahora nuestro usuario verá este curso en su listado de cursos

Universidad Nacional de Quilmes Courses Discover New Help LaraOriana

My Courses

Universidad Nacional de Quilmes Laboratorio de Sistemas Operativos y Redes Unqui - LAB02

About Blog Contact Donate

POWERED BY OPEN edX

© Universidad Nacional de Quilmes. All rights reserved except where noted. edX, Open edX and their respective logos are registered trademarks of edX Inc. Terms of Service & Honor Code - Privacy Policy - Take free online courses at edX.org

Como vimos, el curso no tiene mucha información, por lo que entraremos al cms para modificarlo.

Cerraremos sesión y nos logueamos con el usuario admin para administrarlo

Para eso, iremos a <http://127.0.0.1:18010>, el CMS

Este nos redirigirá a un formulario de autenticación, y luego al <http://127.0.0.1:2001>, que es la URL del micro front end que instalamos, en donde encontramos el estudio para la creación de cursos.

Desde acá podemos editar y crear contenidos para los cursos.

admin

Studio home + New course + New library

Courses Libraries Taxonomies

Search All courses Name A-Z Showing 1 of 1

Laboratorio de Sistemas Operativos y Redes Unqui / LAB02 / LAB002

New to Your Platform Studio? Click "Looking for help with Studio" at the bottom of the page to access our continually updated documentation and other Studio resources. Getting started with Your Platform Studio

Unqui LABO2 Laboratorio de Sistema... Content Settings Tools admin

Content + New section Reindex Collapse all View live

Course outline

Start date Pacing type Checklists Course highlight emails Course tags

Set Date Instructor-paced 2/7 completed Enable now Learn more 0 Manage tags

Modulo 1

- Section highlights
 - Leccion 1
 - Actividad 1

+ New unit

+ New subsection

+ New section

Este es un curso de la materia de Laboratorio de sistemas operativos y redes de la universidad nacional de quilmes


Appears on the course catalog page when students roll over the course name. Limit to ~150 characters

Course overview

↶ ↷ Paragraph **B** *I* U A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [] ^ _ ` ~ { } | ~ < > ...

Introductions, prerequisites, FAQs that are used on your course summary page (formatted in HTML)

Course card image



You can manage this image along with all of your other files and uploads

`/asset-v1:Unqui+LABO2+LABO02+type@asset+block@unquiLogoGrande.png`

Please provide a valid path and name to your course image (Note: only JPEG or PNG format supported)

Course Introduction video

Creating your course organization

You add sections, subsections, and units directly in the outline.

Create a section, then add subsections and units. Open a unit to add course components.

Reorganizing your course

Drag sections, subsections, and units to new locations in the outline.

[Learn more about the course outline](#)

Setting release dates and grading policies

Select the Configure icon for a section or subsection to set its release date. When you configure a subsection, you can also set the grading policy and due date.

[Learn more about grading policy settings](#)

Changing the content learners see

To publish draft content, select the Publish icon for a

En esta sección se puede modificar todo lo relacionado a un curso.

POSIBLES PROBLEMÁTICAS

Conflictos en dependencias de Python

Debido a qué la versión creada del ambiente virtual no era la requerida por el proyecto. Se resolvió instalando pyenv y seteando la versión 3.11 localmente, y con esa versión crear el venv.

```
Collecting click==8.3.1 (from -r requirements/edx/development.txt (line 238))
  Using cached click-8.3.1-py3-none-any.whl.metadata (2.6 kB)
ERROR: Cannot install -r requirements/edx/development.txt (line 1634), -r requirements/edx/development.txt (line 1642), -r requirements/edx/development.txt (line 2024) and astroid==3.3.11 because these package versions have conflicting dependencies.

The conflict is caused by:
  The user requested astroid==3.3.11
  pylint 3.3.9 depends on astroid<=3.4.0.dev0 and >=3.3.8
  pylint-celery 0.3 depends on astroid>=1.0
  sphinx-autoapi 3.6.1 depends on astroid~=4.0; python_version >= "3.12"

Additionally, some packages in these conflicts have no matching distributions available for your environment:
  astroid

To fix this you could try to:
1. loosen the range of package versions you've specified
2. remove package versions to allow pip to attempt to solve the dependency conflict
```

Error en path al qué el proyecto va a buscar los assets estáticos

Hay una variable llamada STATIC_ROOT que es dónde deberán estar localizados los archivos generados.

El archivo que buscara es webpack-stats.json. Se puede mover este archivo a la ubicación esperada, o cambiar la configuración con el path del directorio deseado.

```
OSEError at /
Error reading /home/orianita/Desktop/staticfiles/webpack-stats.json. Are you sure webpack has generated the file and the path is correct?

Request Method: GET
Request URL: http://127.0.0.1:18000/
Django Version: 5.2.9
Exception Type: OSEError
Exception Value: Error reading /home/orianita/Desktop/staticfiles/webpack-stats.json. Are you sure webpack has generated the file and the path is correct?
Exception Location: /home/orianita/Desktop/edx-platform/venv/lib/python3.11/site-packages/webpack_loader/loaders.py, line 58, in load_assets
Raised during: lms.djangoapps.branding.views.index
Python Executable: /home/orianita/Desktop/edx-platform/venv/bin/python
Python Version: 3.11.14
Python Path: ['/home/orianita/Desktop/edx-platform',
              '/home/orianita/Desktop/edx-platform',
              '/home/orianita/.pyenv/versions/3.11.14/lib/python311.zip',
              '/home/orianita/.pyenv/versions/3.11.14/lib/python3.11',
              '/home/orianita/.pyenv/versions/3.11.14/lib/python3.11/lib-dynload',
              '/home/orianita/Desktop/edx-platform/venv/lib/python3.11/site-packages',
              '/home/orianita/Desktop/edx-platform/venv/lib/python3.11/site-packages/setuptools/_vendor']
Server time: Sun, 07 Dec 2025 23:50:28 +0000

Traceback Switch to copy-and-paste view
/home/orianita/Desktop/edx-platform/venv/lib/python3.11/site-packages/webpack_loader/loaders.py, line 55, in load_assets
55.         with open(self.config["STATS_FILE"], encoding="utf-8") as f:
           ~~~~~^~~~~~
▶ Local vars
```

Errores de missing django app label.

Al correr las migraciones o levantar la app nos encontramos con que ciertas aplicaciones de django del proyecto se estaban llamando antes de ser cargadas, por lo que se mostraba un runtime error.

Para solucionar esto pusimos en los minimal.yml (archivos de configuración) una variable llamada INSTALLED_APPS que es la que usa django para cargar las aplicaciones o dependencias.

A medida que fuimos avanzando con la instalación, empezamos a encontrarnos con errores del tipo "**Application labels aren't unique, duplicates: ...**" que básicamente nos indica que hay aplicaciones registradas más de una vez, justamente las que incluimos a mano en el archivo de configuración, por lo tanto las sacamos. Luego de esto no tuvimos más problemas en relación a las apps instaladas.

Errores generales de configuración

- Autenticación en mongoDB y MySQL,
- Client id de la configuración de cms no coincide con el client id de la app de Oauth,
- Configuraciones faltantes de cookies de sesión y de caché,
- Configuraciones de URLs de micro front end

Estos problemas fueron resueltos agregando variables faltantes al archivo de configuración.

Como avisa el readme del repositorio, hay muchas de estas cosas que no están documentadas ya que la versión principal y más mantenida es Tutor, que está dockerizada y evita muchos de estos problemas de configuración.

Para algunos de estos errores, como el del client_id de cms y Oauth, utilizamos la inteligencia artificial de cursor, ya que muchos de los errores relacionados a configuración no eran muy explícitos en que es lo que estaba saliendo mal.

Para otros problemas, como los errores en las cookies de sesión y caché, por ejemplo sirvieron publicaciones de la comunidad como [esta](#), que si bien no tenían el mismo problema explícito, nos sirvió para saber que existían configuraciones para las cookies.

EJEMPLO DE ARCHIVO DE CONFIGURACIONES

CMS

```
DATA_DIR: "/tmp/edx-platform/data_dir/"

# For just the CMS
LMS_ROOT_URL: "http://127.0.0.1:18000"
LMS_INTERNAL_ROOT_URL: "http://127.0.0.1:18000"
CMS_BASE: "127.0.0.1:18010"
CMS_ROOT_URL: "http://127.0.0.1:18010"
EDX_ROOT_URL: "http://127.0.0.1:18010"

API_ACCESS_MANAGER_EMAIL: "api-access@example.com"

# Platform Branding
PLATFORM_NAME: "Universidad Nacional de Quilmes"

DEBUG: true

# Static files configuration (for development)
STATIC_URL: "/static/studio/"
# STATIC_ROOT will use default from common.py if not set
```

LMS

```
MEDIA_ROOT: "/tmp/edx-platform/media/"

DATA_DIR: "/tmp/edx-platform/data_dir/"

# For just the LMS
LMS_ROOT_URL: "http://127.0.0.1:18000"
LMS_INTERNAL_ROOT_URL: "http://127.0.0.1:18000"
EDX_ROOT_URL: "http://127.0.0.1:18000"

LEARNING_MICROFRONTEND_URL: "http://127.0.0.1:2000"

API_ACCESS_MANAGER_EMAIL: "api-access@example.com"

PLATFORM_NAME: "Universidad Nacional de Quilmes"
PLATFORM_DESCRIPTION: "Universidad Nacional de Quilmes"

DEBUG: true

# Static files configuration (for development)
STATIC_URL: "/static/"
# STATIC_ROOT will use default from common.py if not set
```