

Sistemas Software Basados en Web

Prácticas

Autor

Abdullah Taher Saadoon AL-Musawi



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

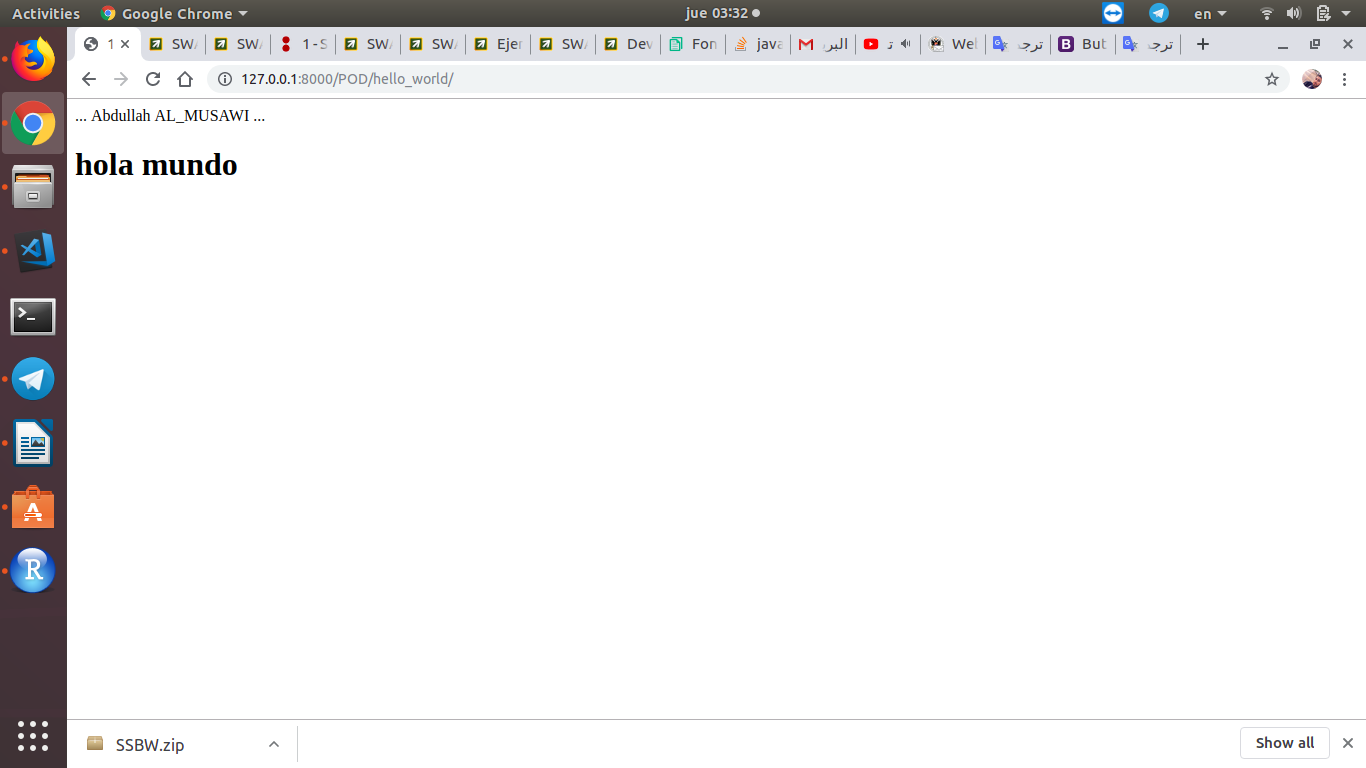
—

Granada, junio de 2019

The first exercises that I have sent it which was Development environment with docker-compose with 6 simple examples by python

1- hello world

with link: <http://127.0.0.1:8000/POD/hello_world/>



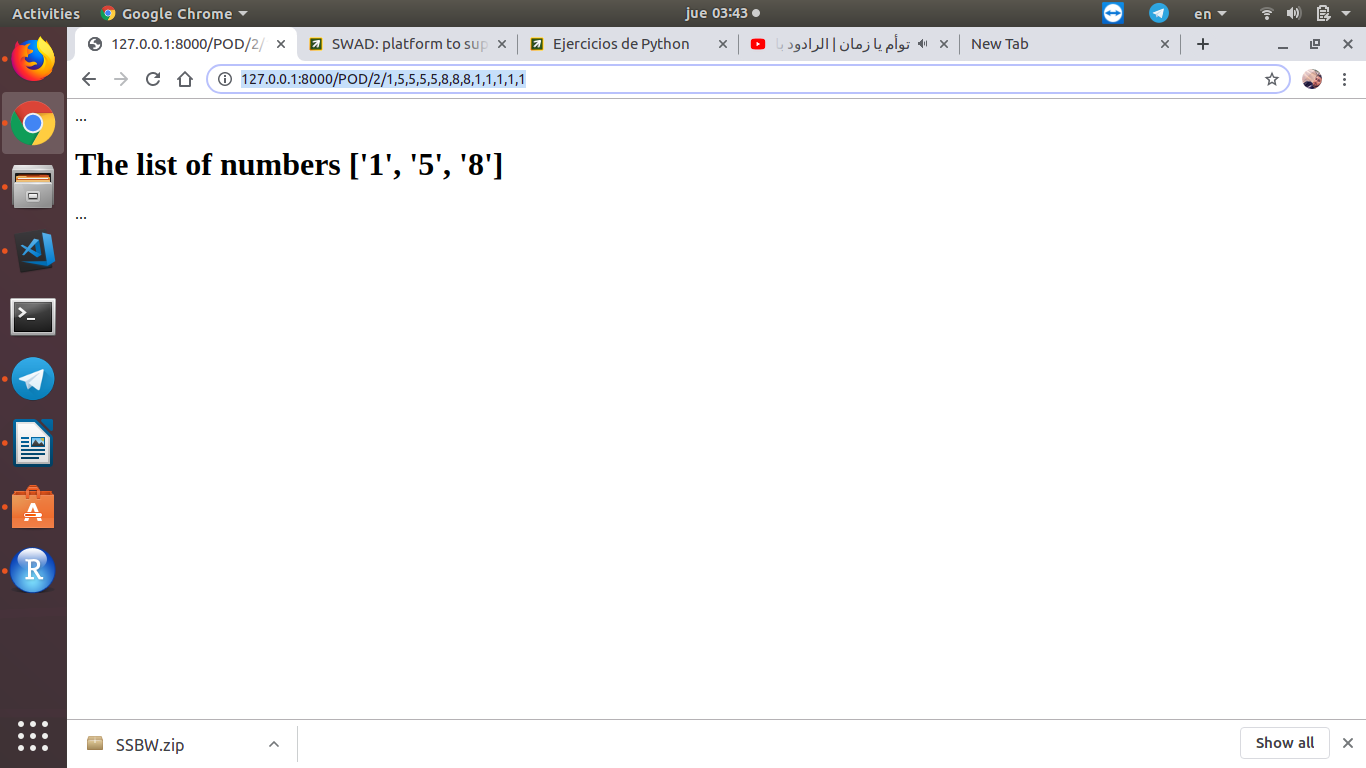
2- Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last chars of the string are the same. Note: python does not have a ++ operator, but += works.

With link: [http://127.0.0.1:8000/POD/1/[list-of-strings](http://127.0.0.1:8000/POD/1/%5Blist-of-strings)]



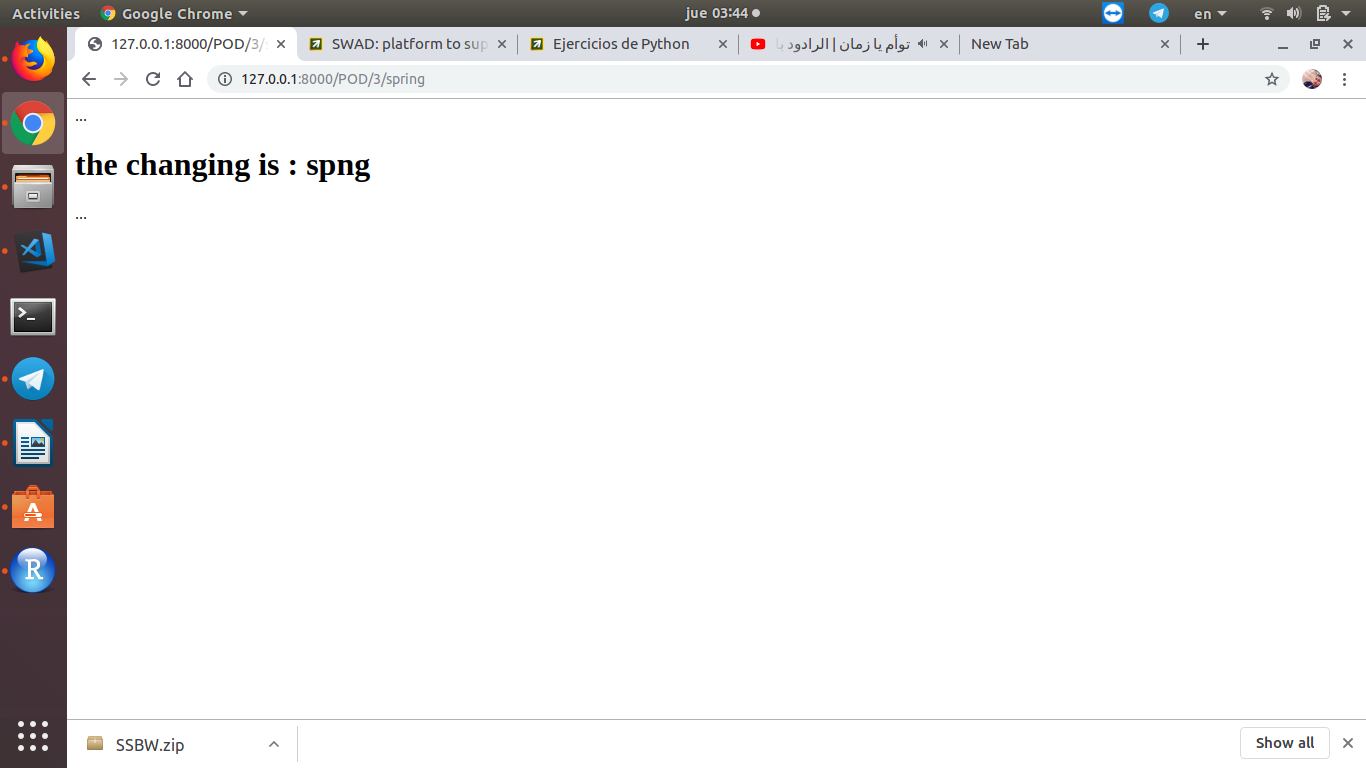
3- Given a list of numbers, return a list where all adjacent == elements have been reduced to a single element, so [1, 2, 2, 3] returns [1, 2, 3]. You may create a new list or modify the passed in list.

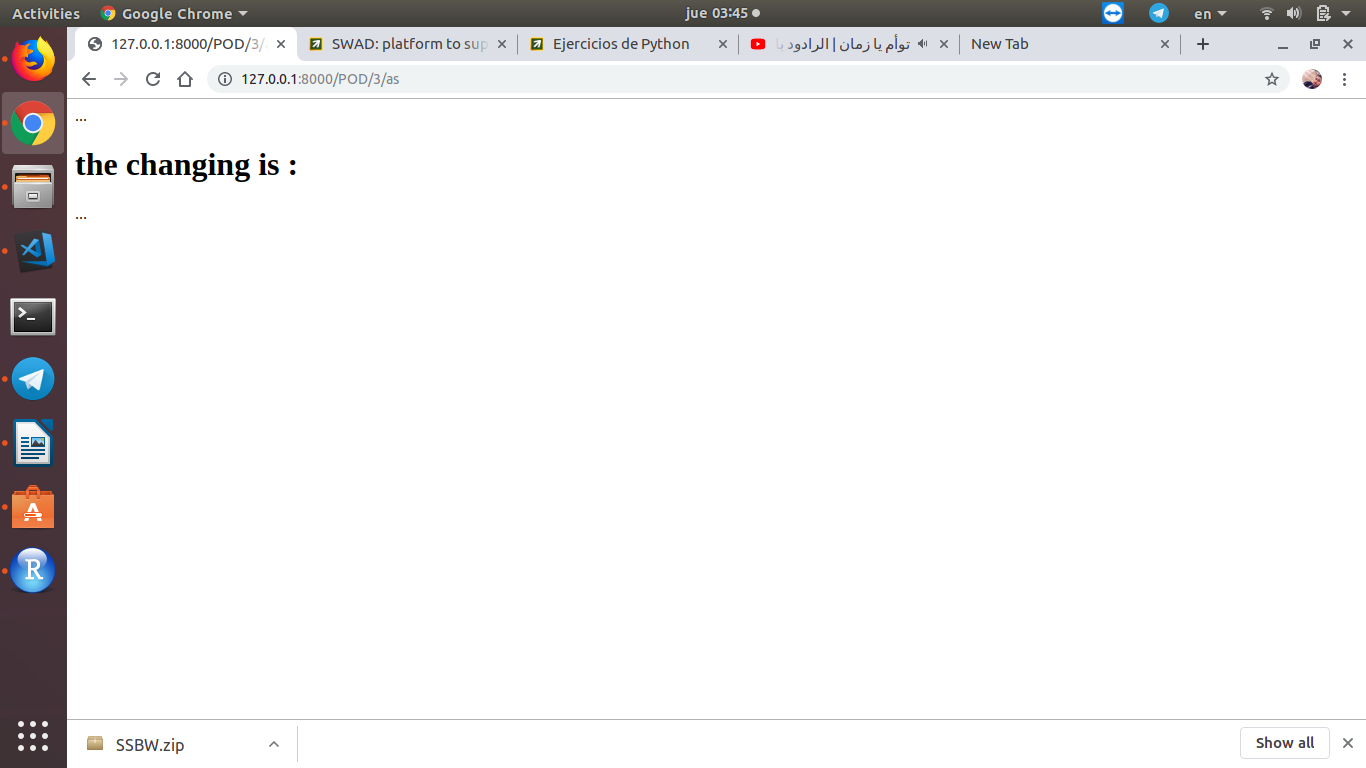
With link: [http://127.0.0.1:8000/POD/2/[list-of-numbers](http://127.0.0.1:8000/POD/2/%5Blist-of-numbers)]



4- Given a string s, return a string made of the first 2 and the last 2 chars of the original string, so 'spring' yields 'spng'. However, if the string length is less than 2, return instead the empty string.

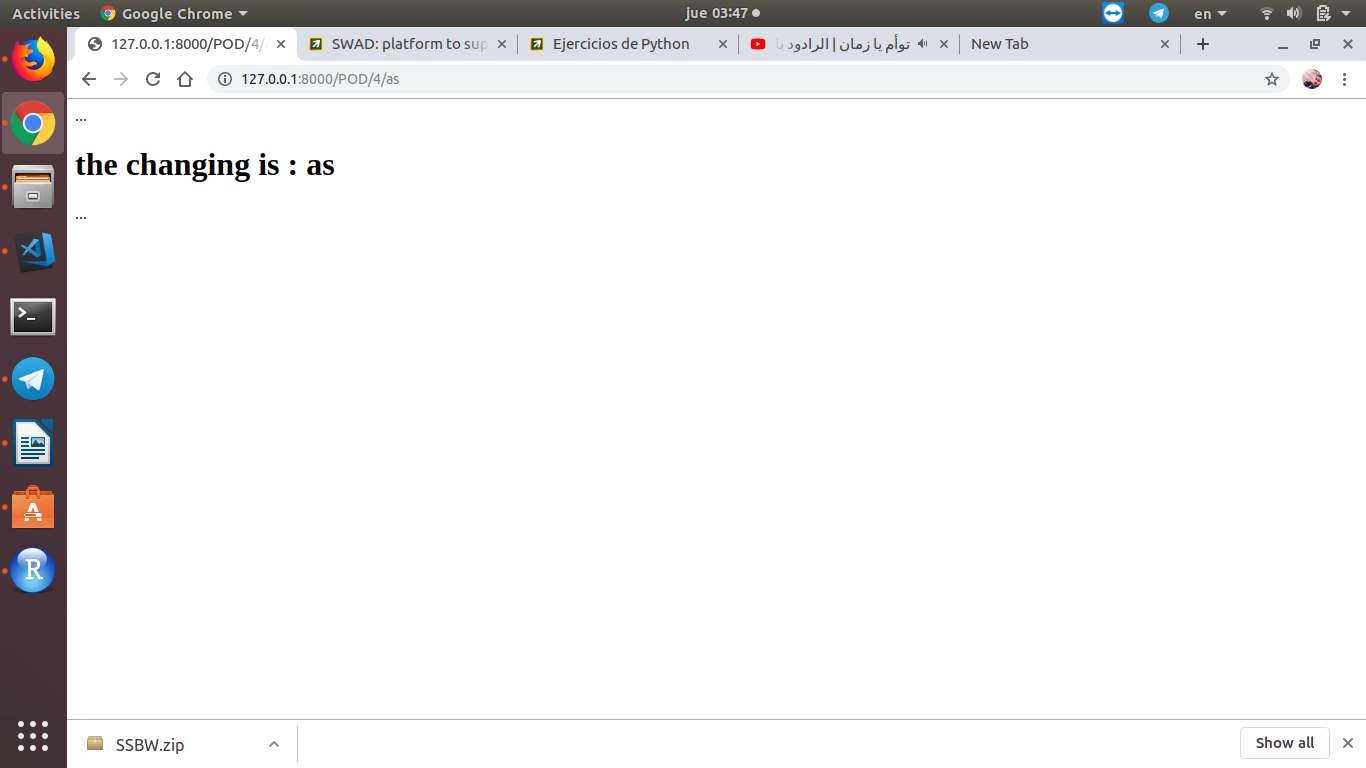
With link: <http://127.0.0.1:8000/POD/3/String>

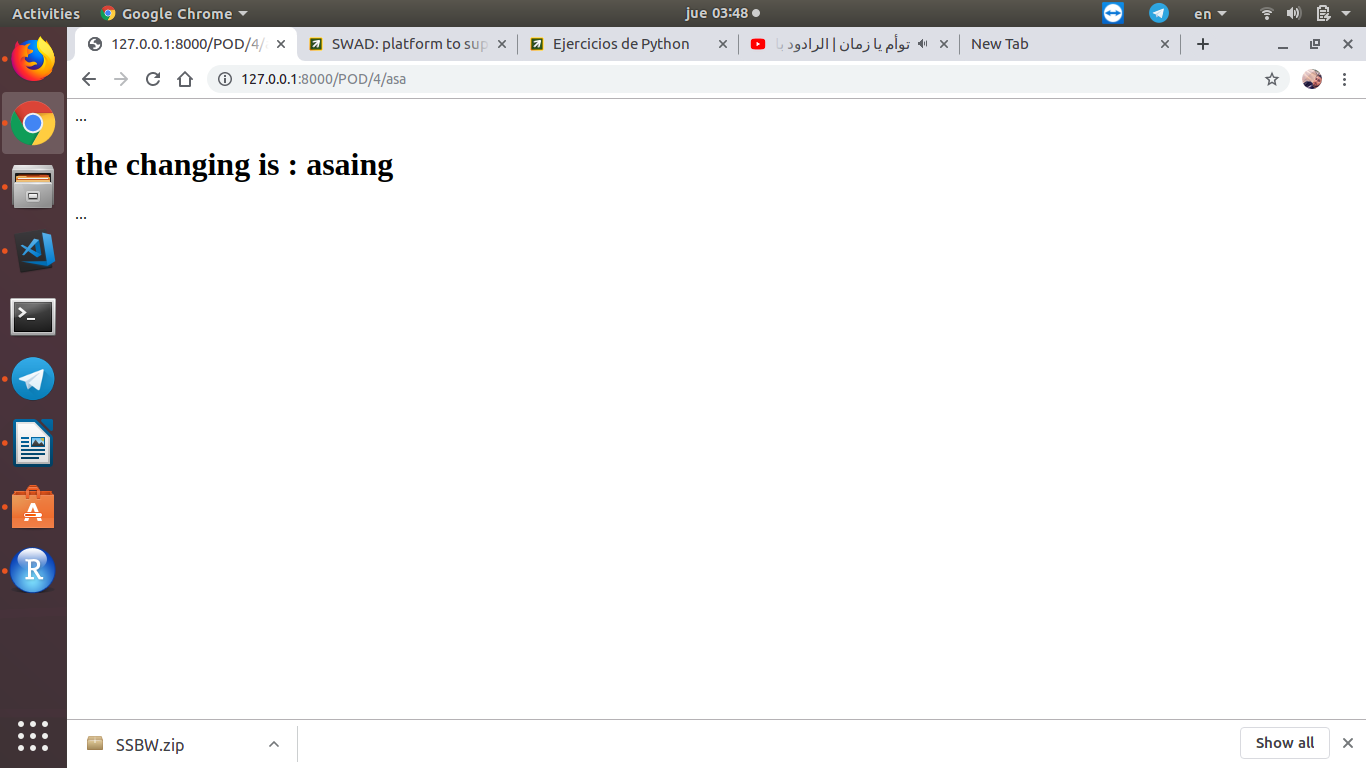


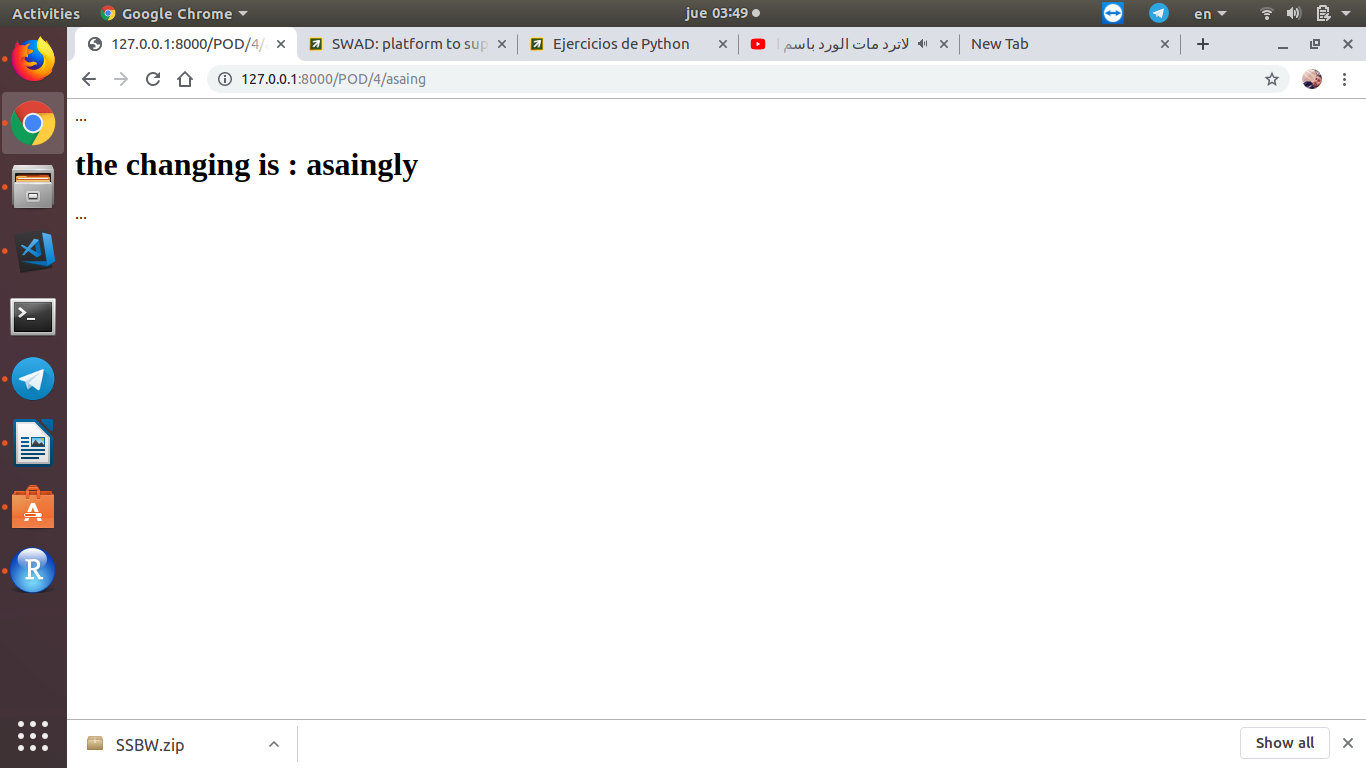


5- Given a string, if its length is at least 3, add 'ing' to its end. Unless it already ends in 'ing', in which case add 'ly' instead. If the string length is less than 3, leave it unchanged.

With link: <http://127.0.0.1:8000/POD/4/String>

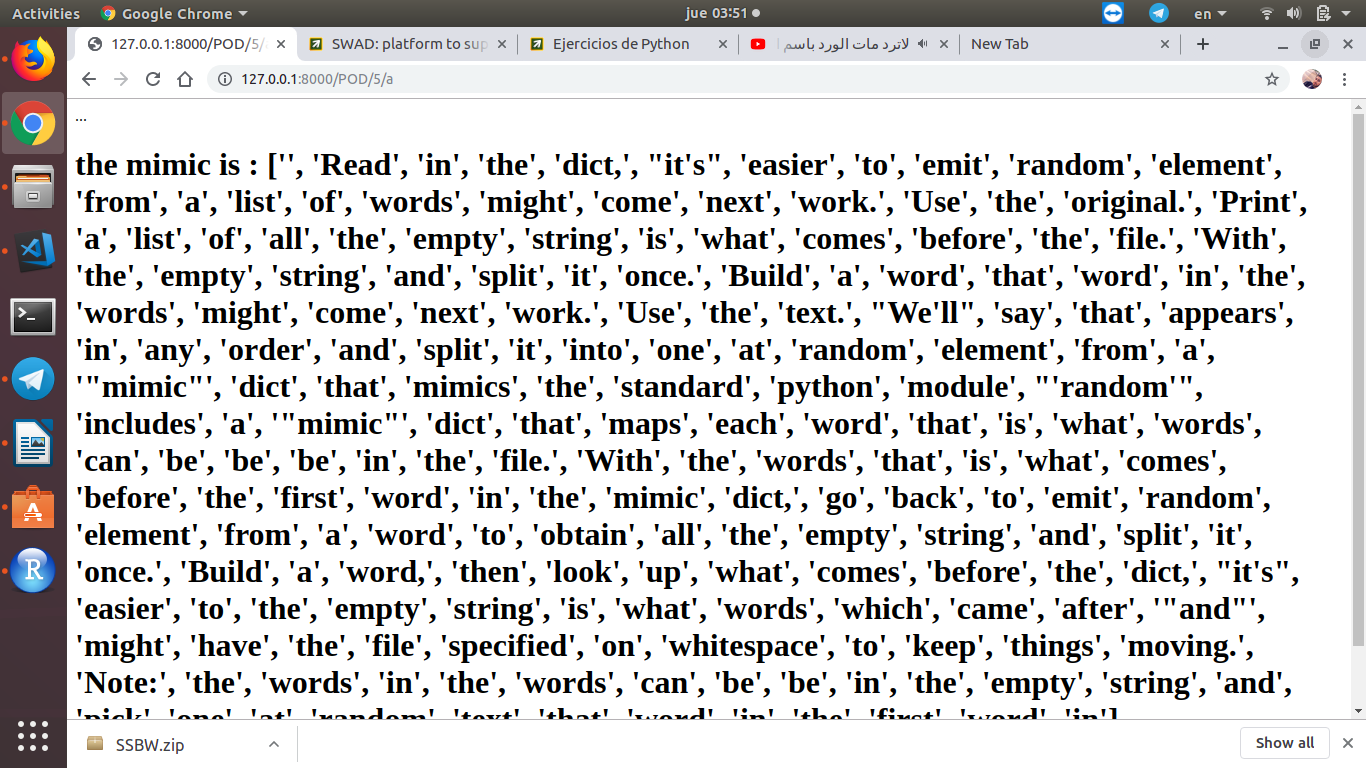






6- Read any text file specified on the command line. Do a simple split() on whitespace to obtain all the words in the file. Rather than read the file line by line, it's easier to read it into one giant string and split it once.

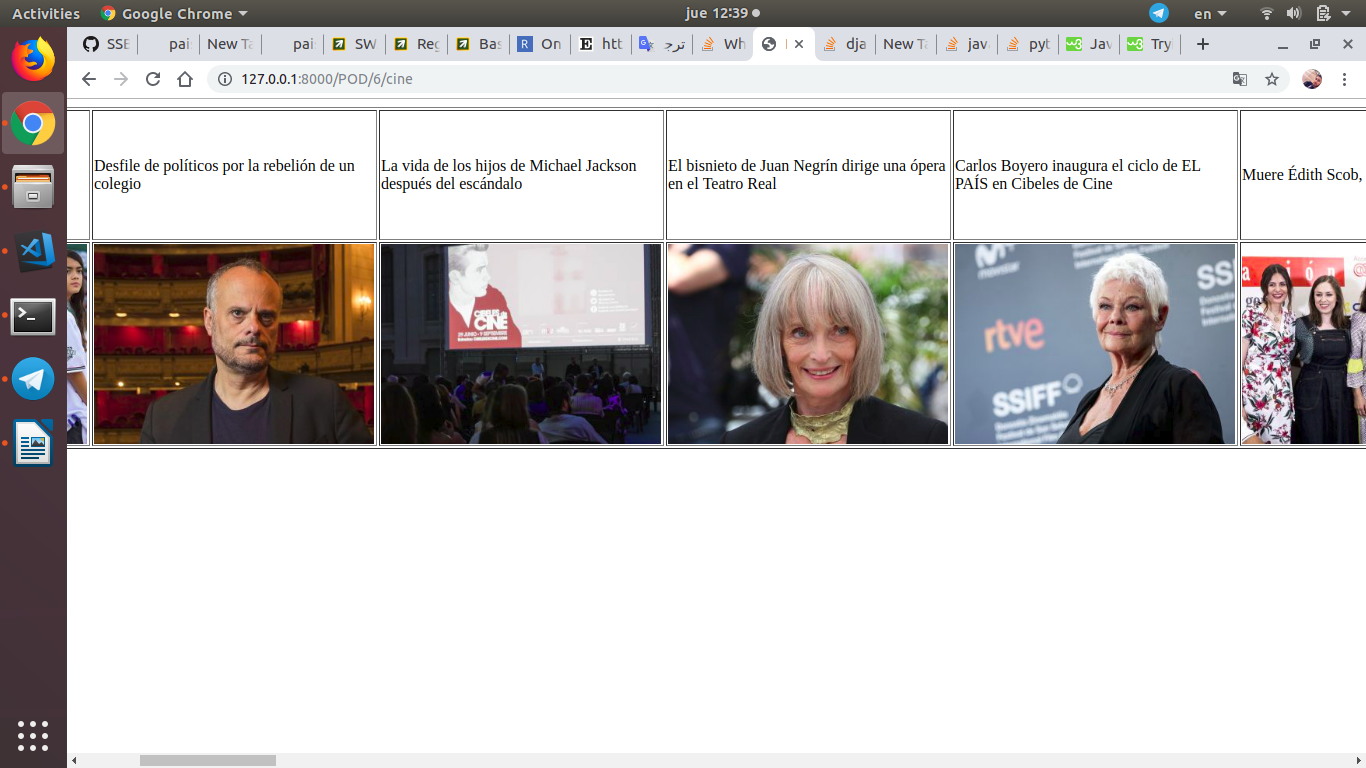
With link: <http://127.0.0.1:8000/POD/5/String>



tarea3

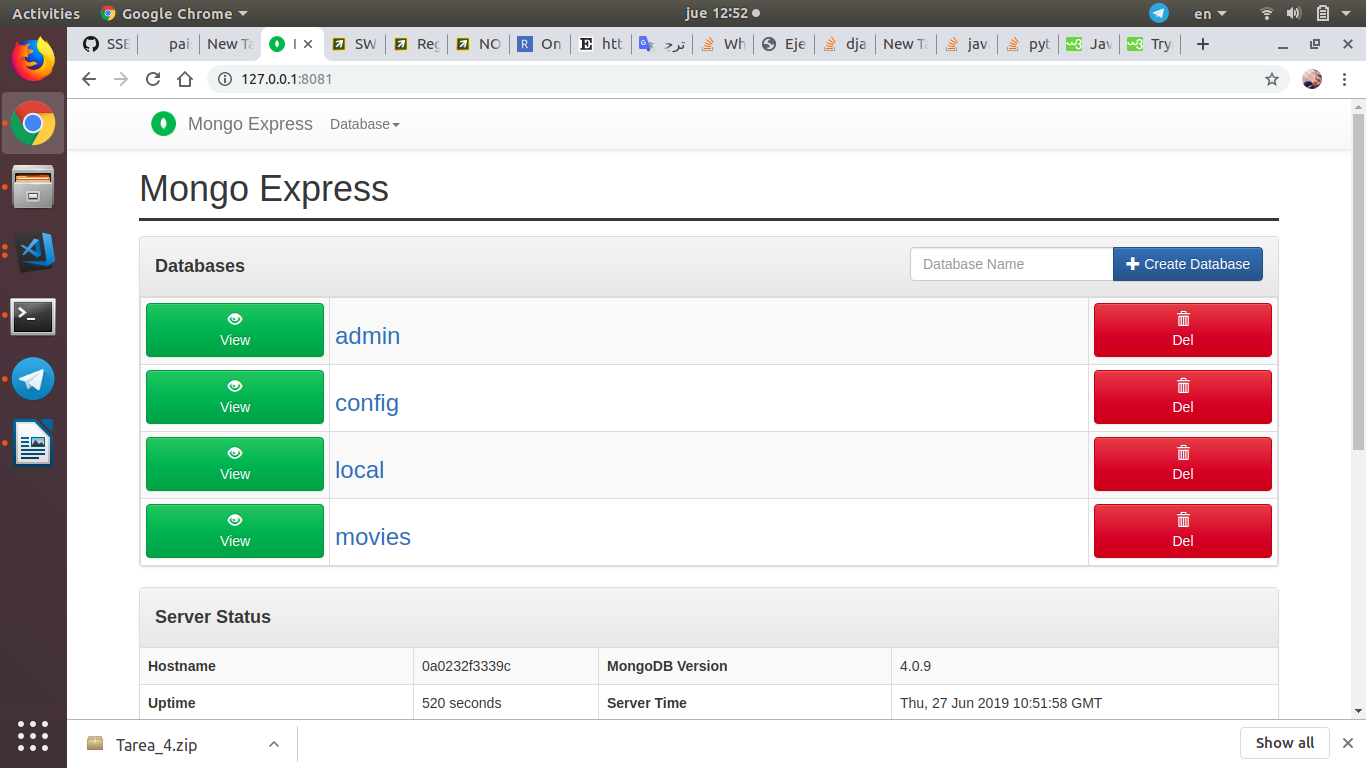
exercise in Basic Python Exercises of Regular Expressions, so I did an application with Regular Expressions which does get request from <https://servicios.elpais.com/rss/> + title (for example ‘cine’) the data will be as xml, so we have to get title and image from this xml, and show them in the html file

the result With link: <http://127.0.0.1:8000/POD/6/String>

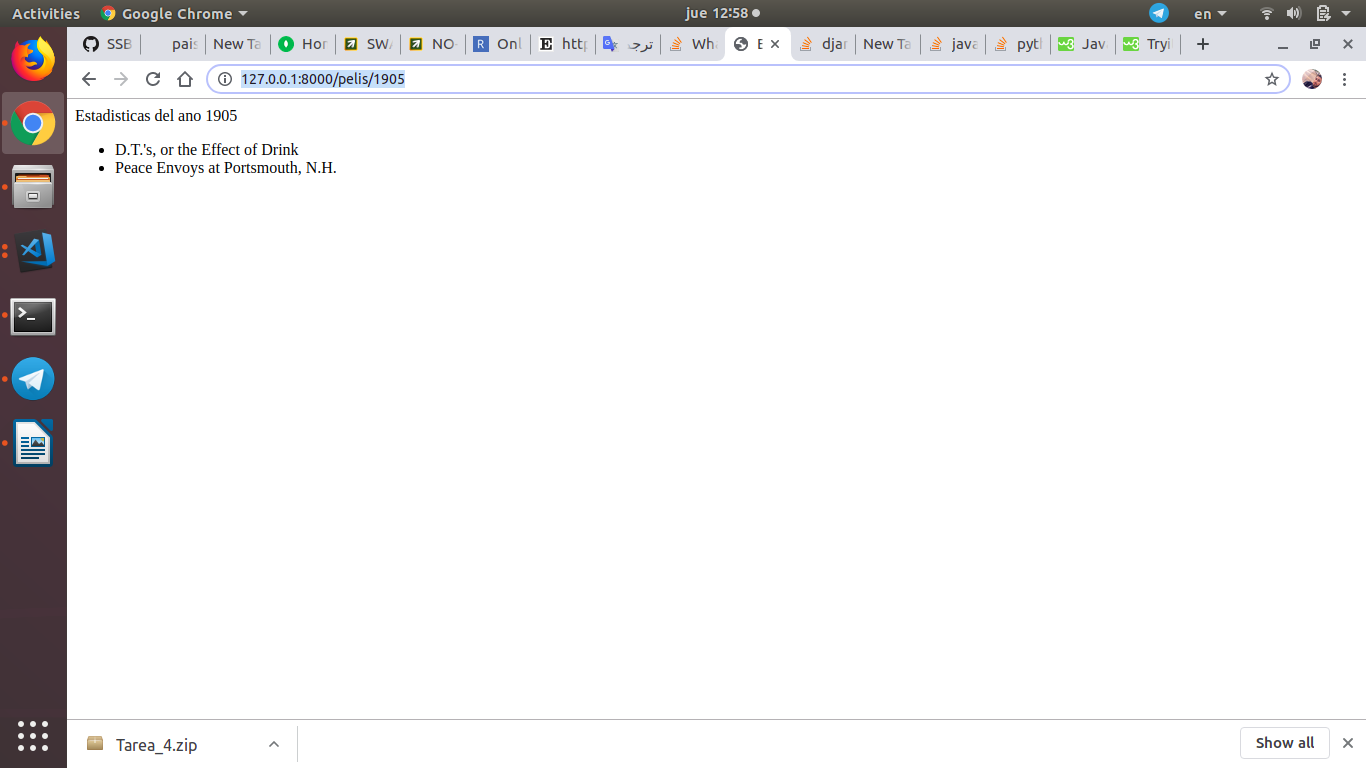


tarea4

we have installed and use the non-sql database MongoDB, then We have created a service for mongo dB, we can see our database by GUI in our browsers following this link: <http://127.0.0.1:8081/>



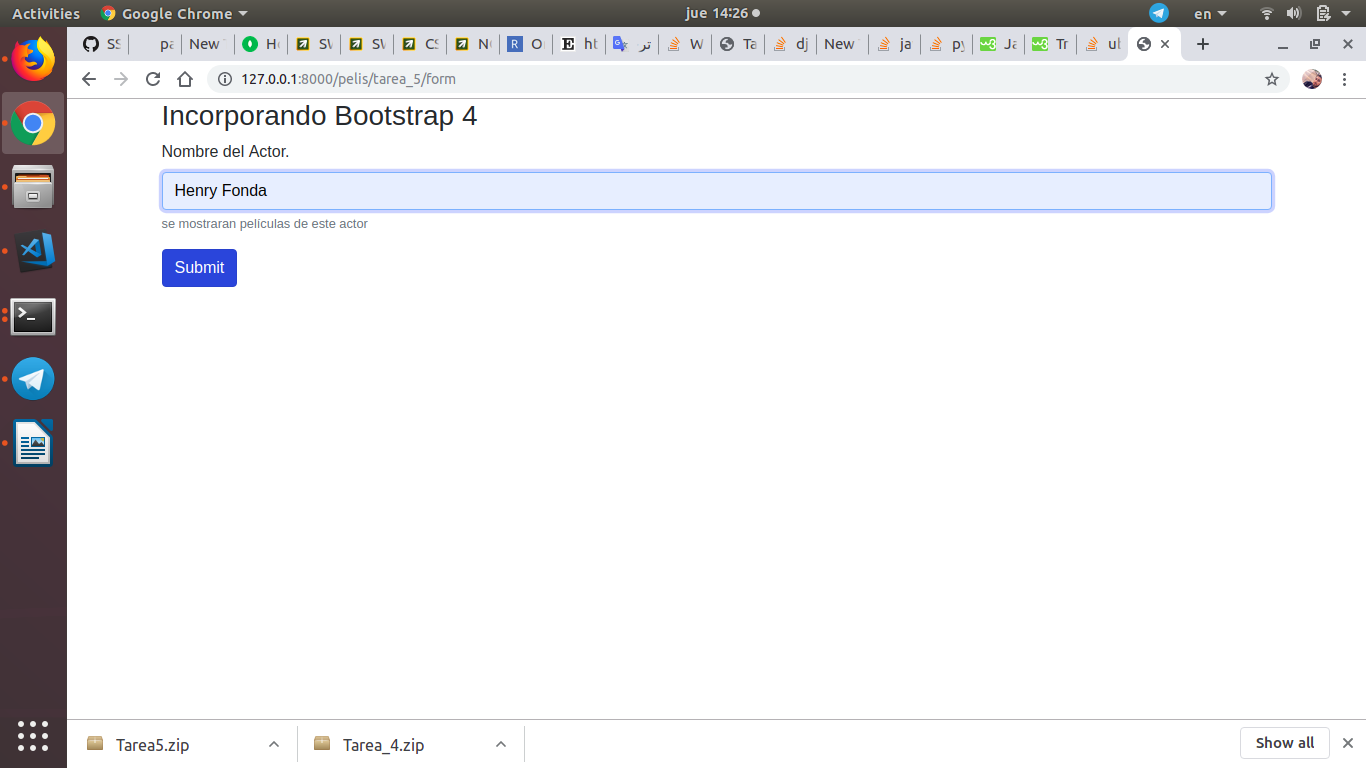
We downloaded the data of pelis movies and import them to the DB with the utility mongo import, so by simple method we can show the titles of movies by its year creating follow this link: [http://127.0.0.1:8000/pelis/[year](http://127.0.0.1:8000/pelis/%5Byear)]

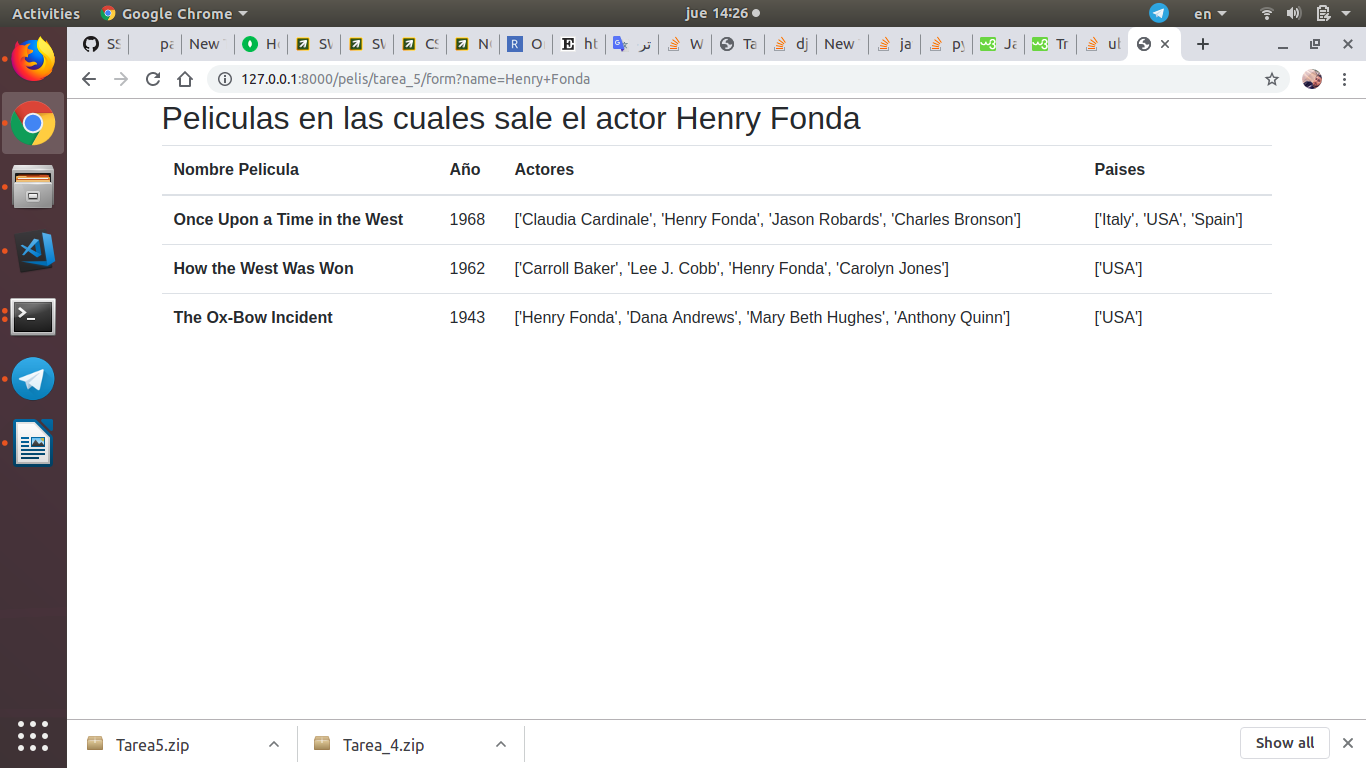


tarea5

in this practice we have incorporated some of the CSS frameworks 'responsive' to the previous task. and we have incorporated the model with mongo engine, and we have made a similar query using a [form](https://developer.mozilla.org/es/docs/Learn/HTML/Forms/Your_first_HTML_form) .

The link: <http://127.0.0.1:8000/pelis/tarea_5/form>

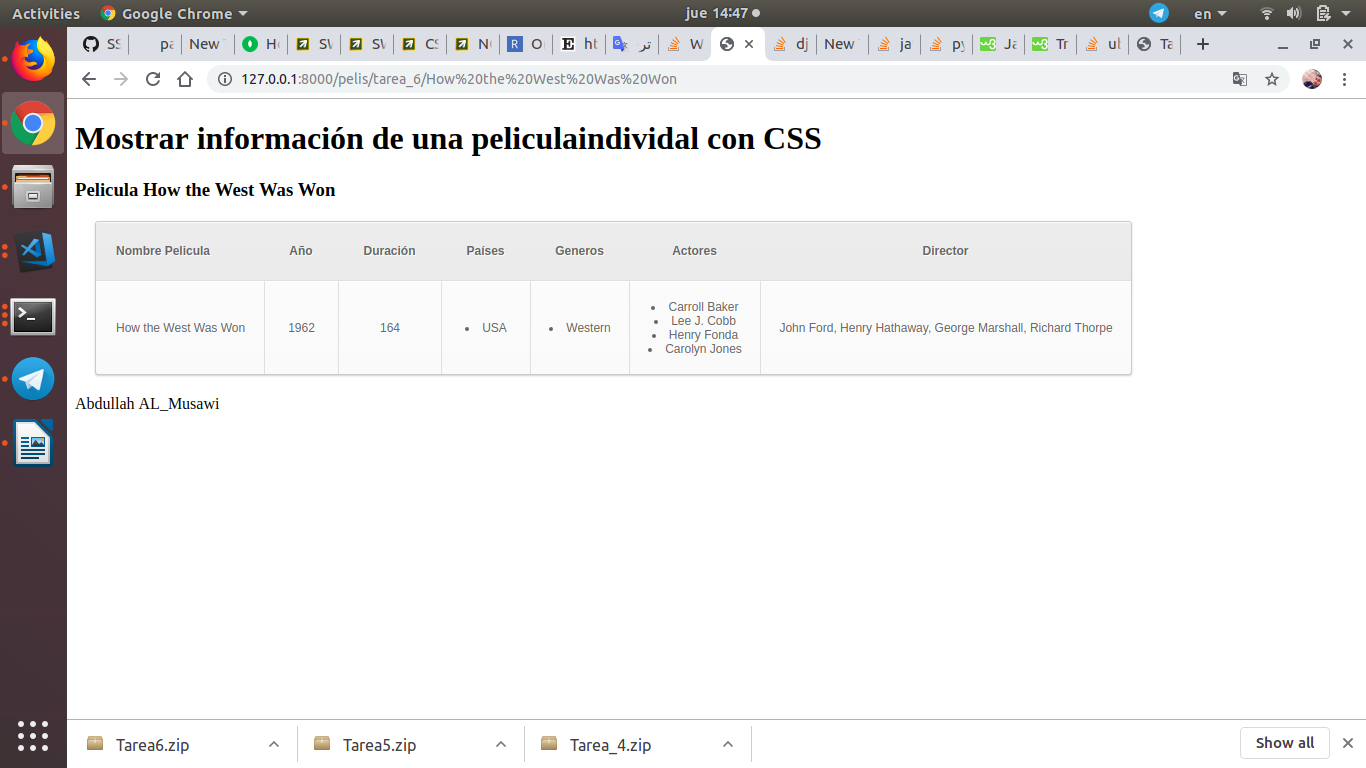




tarea6

It is this task we have made a web page to present the information of an individual movie. we have done it directly coding CSS

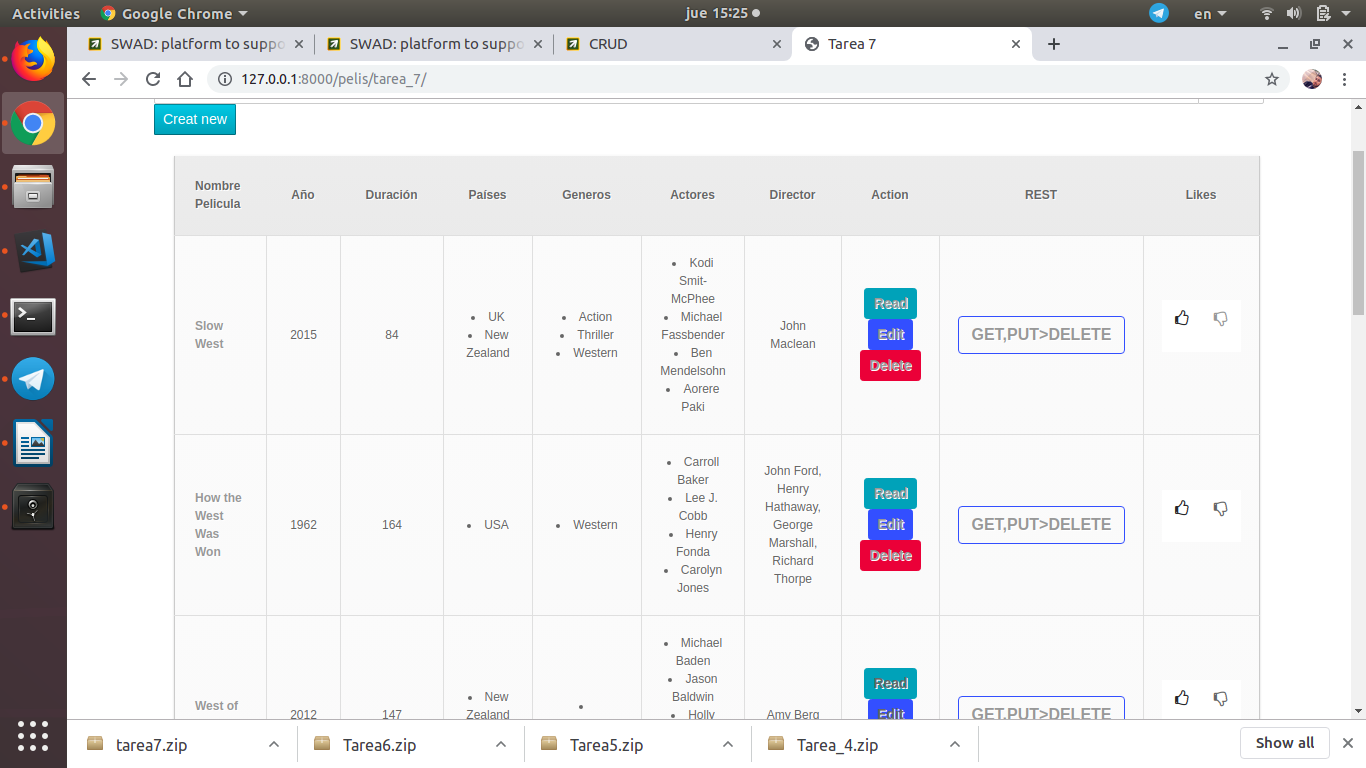
the Link: <http://127.0.0.1:8000/pelis/tarea_6/MovieName>



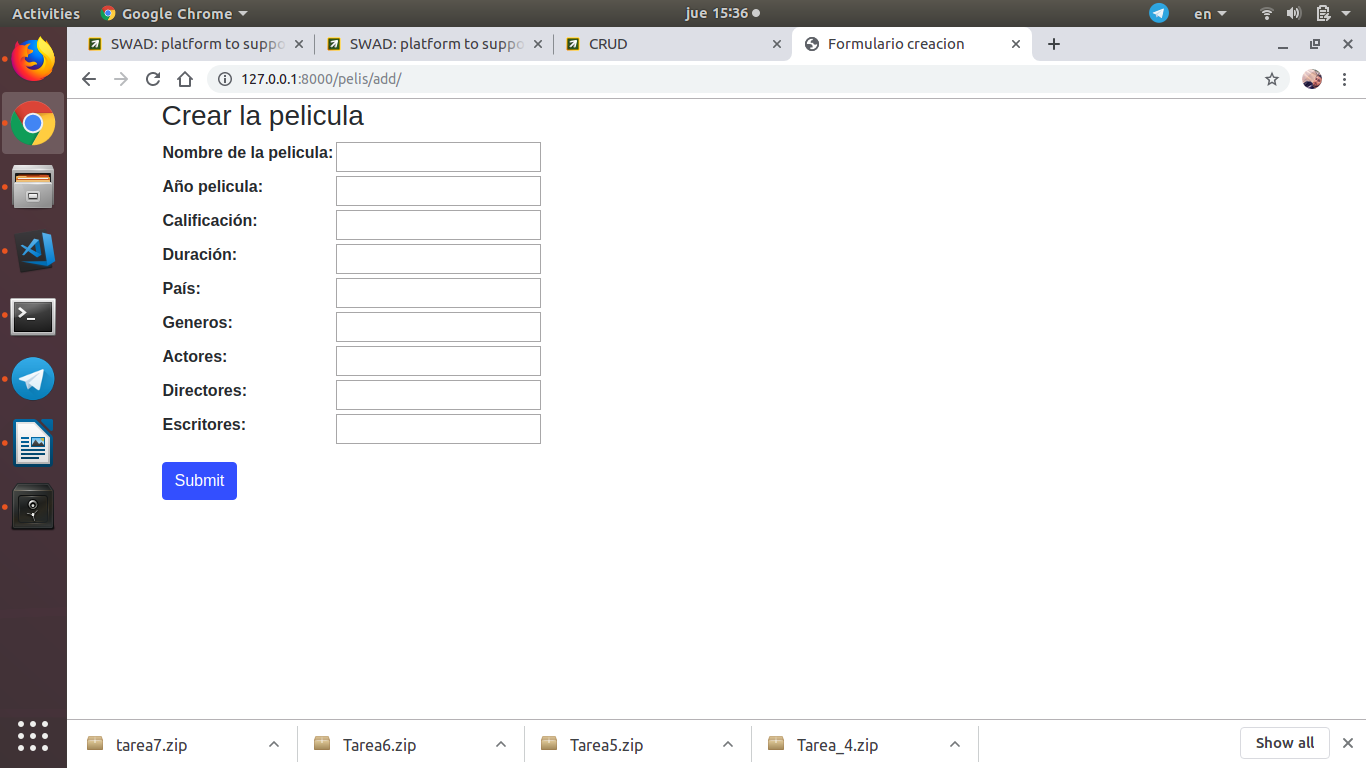
tarea7

In this task we made the rest of the CRUD of the application, using the Django [forms](https://docs.djangoproject.com/en/2.1/topics/forms/) class.

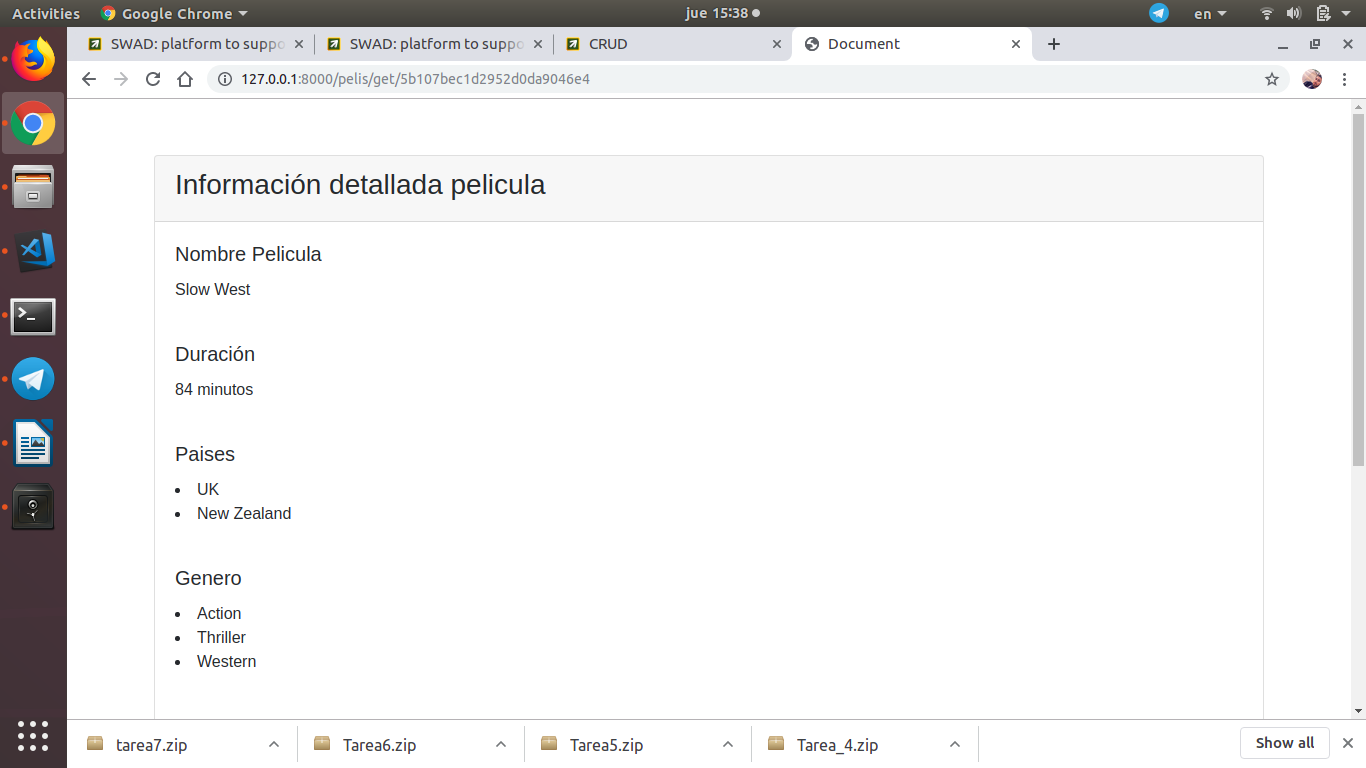
GET all Movies: with The Link: <http://127.0.0.1:8000/pelis/tarea_7/>



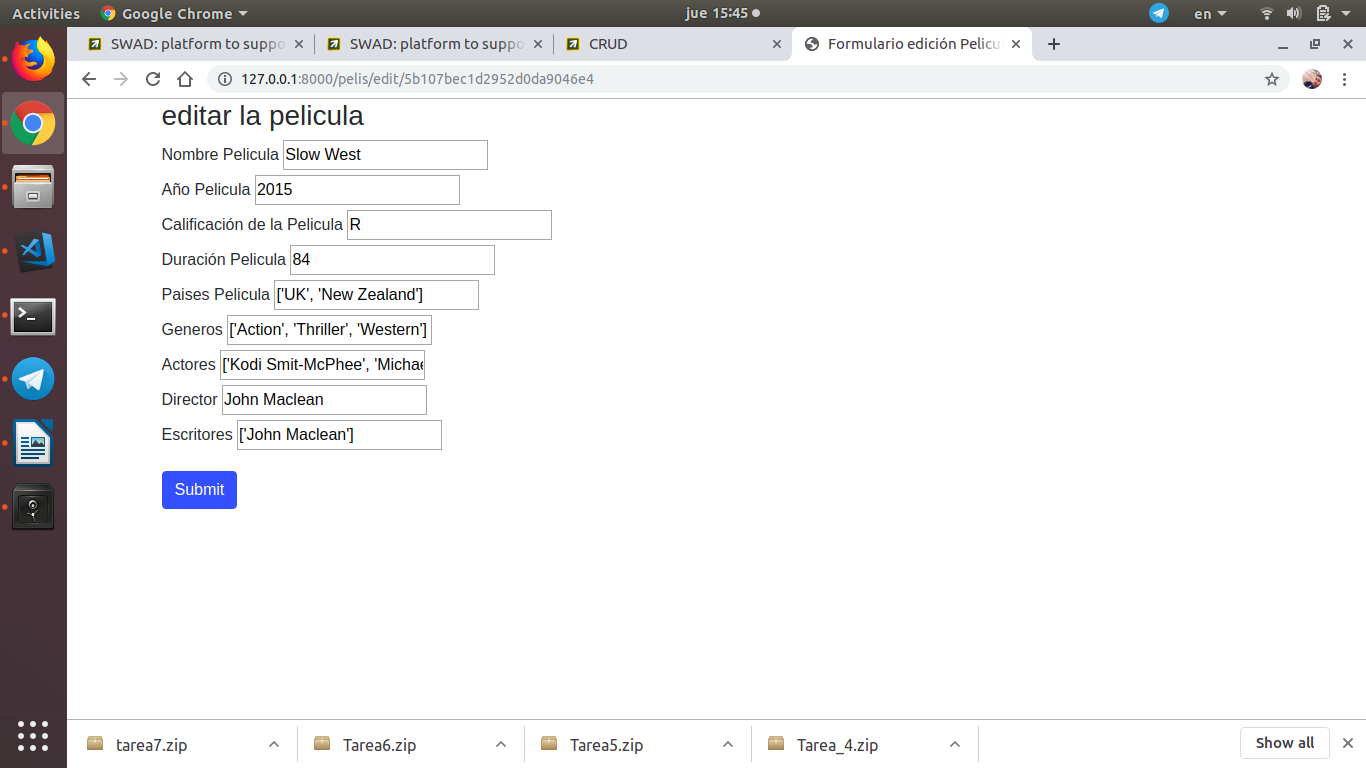
add new Movies(Post):by click create new, the Link: <http://127.0.0.1:8000/pelis/add/>



show movie info. (get): by click on read button, the link: <http://127.0.0.1:8000/pelis/get/MovieID>



edit a movie (Put):by click on Edit button, with the Link: <http://127.0.0.1:8000/pelis/edit/MovieID>

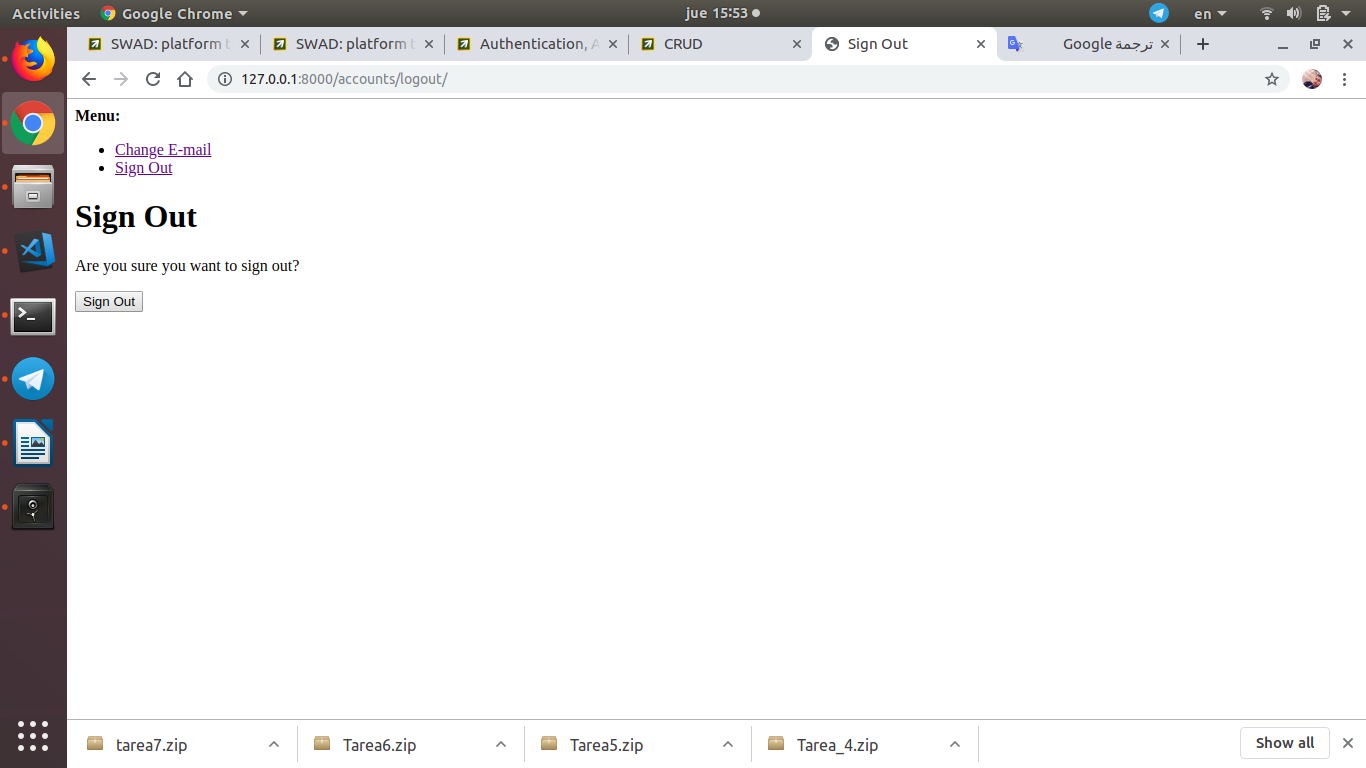
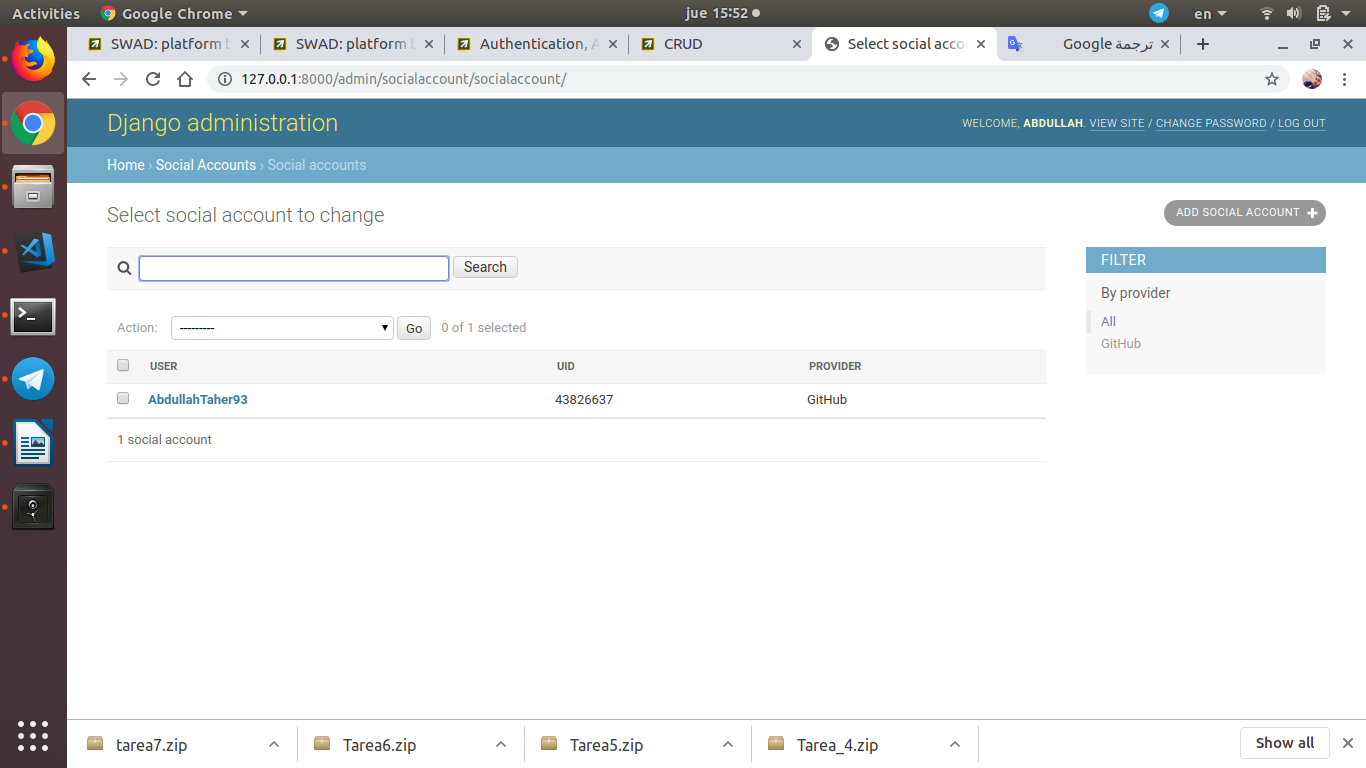


Delete a Movie: by click delete button, with link <http://127.0.0.1:8000/pelis/delete/MovieID>.

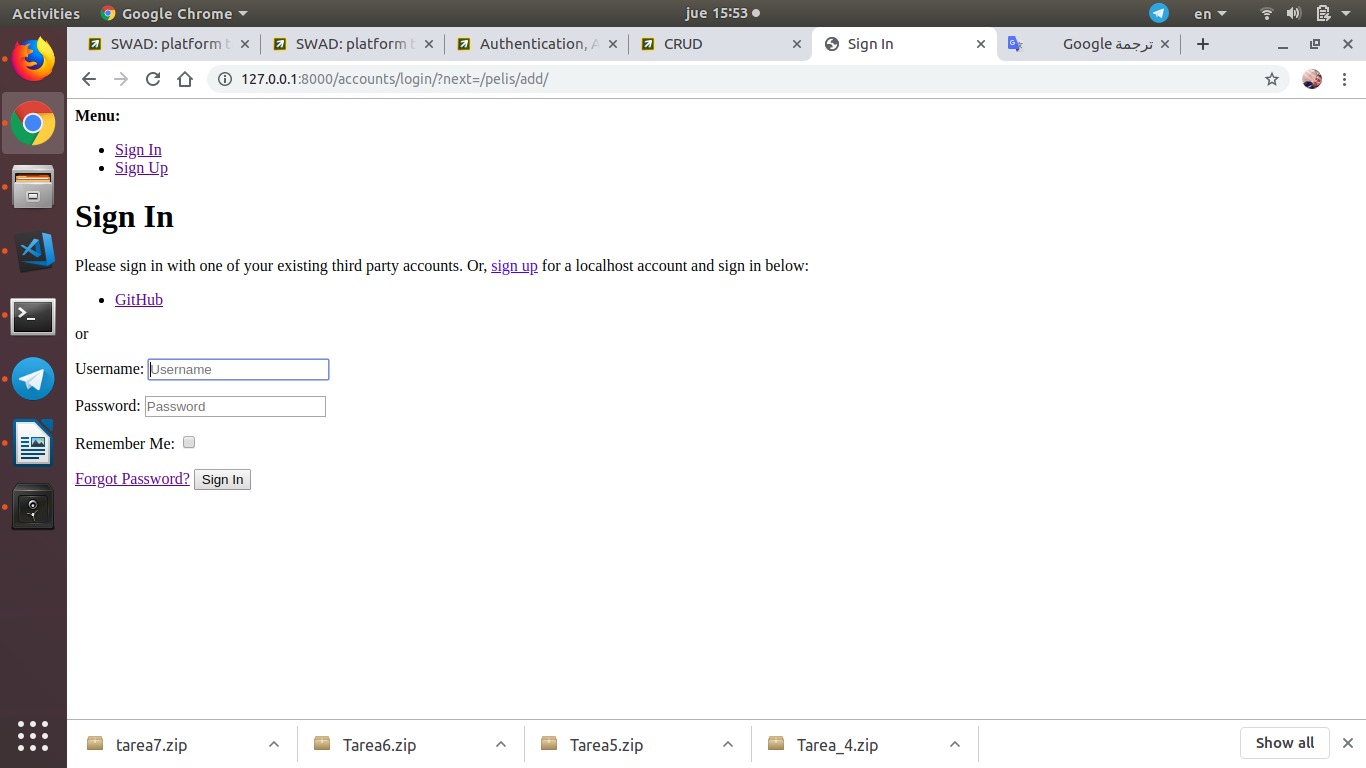
Tarea8

in this practice we have managed the authorization of users in the views, simply including the [decorator](mailto:decorator@login_required) [@login\_required](mailto:decorator@login_required) before each 'view' of Django: [login required decorator](https://docs.djangoproject.com/en/2.2/topics/auth/default/#the-login-required-decorator)

management of the obliviated password with tokens, and delegation of Authentication in social networks using oauth2. We have used GitHub for that



so now when the user clicks on a button of CRUD he will need to authorization, so he can do authorization by GitHub



tarea10

in this task we have added 'night' mode to change the colors of the CSS and typeahead plugin for database searches, that by using JavaScript code through the jQuery library.





Tarea11

We have putted icons ‘Like’, and ‘unlike’, we used Font Awesone, or those suggested by Bootstrap

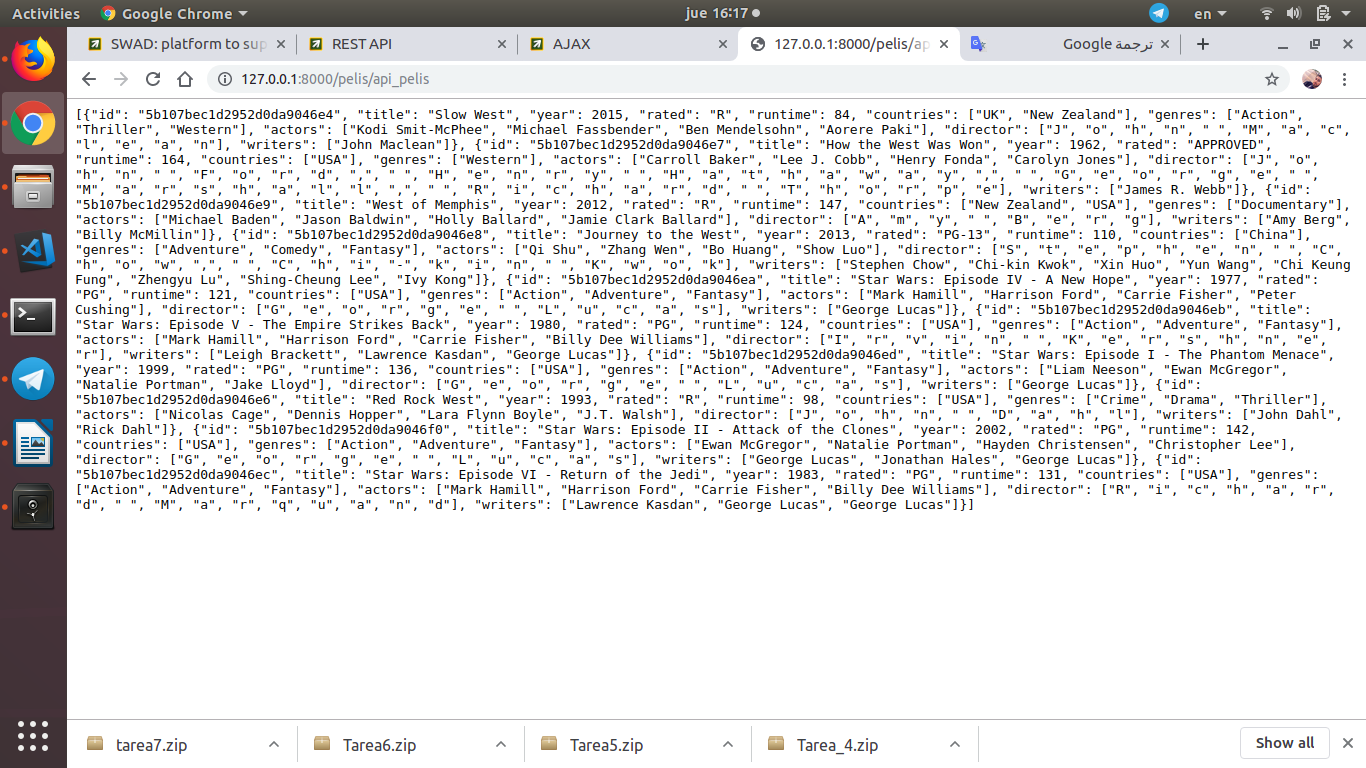


Tarea12

In this task we have added a REST API to our application GET, POST, PUT, DELETE and response the data of movies as a json,

follow this Link: <http://127.0.0.1:8000/pelis/api_pelis>

GET, POST



GET a movie, PUT and DELETE by click on button ‘GET,PUT>DELETE’



follow this Link: <http://127.0.0.1:8000/pelis/api_peli/MovieID>

APIS from viewsets classes

We now followed Django + MongoDB = Django REST Framework Mongo engine to include response from classes that includes DRF. In a new viewsets.py file.

With this Link: <http://127.0.0.1:8000/pelis/apipelis/>

