**SISTEMA DE RECOMENDACIONES**



**Institución: Universidad Franz Tamayo**

**Título del proyecto: SISTEMA DE RECOMENDACIONES**

**Nombre del estudiante: Mel Dusan Mejia Garcia**

**Nombres docentes: Miguel Paco**

Contenido

[ANTECEDENTES 3](#_Toc38626958)

[OBJETIVO 3](#_Toc38626959)

[OBJETIVOS ESPECIFICOS 3](#_Toc38626960)

[MARCO TEORICO 3](#_Toc38626961)

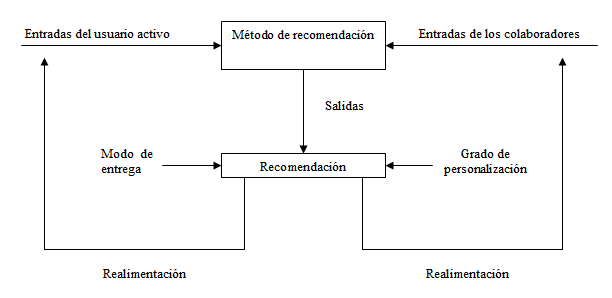
[METODOLOGIA 4](#_Toc38626962)

[DESARROLLO 4](#_Toc38626963)

[CONCLUSIONES 4](#_Toc38626964)

# ANTECEDENTES

Los sistemas de recomendaciones son herramientas que generan recomendaciones sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por los usuarios. El uso de estos sistemas se está poniendo cada vez más de moda en Internet debido a que son muy útiles para evaluar y filtrar la gran cantidad de información disponible en la Web con objeto de asistir a los usuarios en sus procesos de búsqueda y recuperación de información. En este trabajo realizaremos una revisión de las características y aspectos fundamentales relacionados con el diseño, implementación y estructura de los sistemas de recomendaciones.



# OBJETIVO

**Desarrollar** una aplicación informática que ayude predecir los productos que querrá adquirir un usuario en particular.

# OBJETIVOS ESPECIFICOS

1. Analizar la importancia de un sistema de recomendaciones.
2. Desarrollar la aplicación
3. Probar la aplicación (localmente)
4. Desplegar para su uso

# METODOLOGIA

Método Deductivo En el método deductivo se utiliza la lógica y toda la información general para formular y dar una solución posible a un problema dado. Luego se comprueba la solución en varias situaciones normales. Por lo tanto, en el método deductivo, el razonamiento va de lo general a lo específico. Por todo esto el método que utilizaremos será el método deductivo ya que es el que utiliza la información general para dar una solución a un problema dado.

# Técnicas de investigación

Para el desarrollo de este sistema se utilizó el siguiente mecanismo para hacer la recolección de datos. Se recopilo información relacionada sobre los tipos de tecnologías para llevar un sistema de recomendaciones adecuado y también la recopilación de datos.

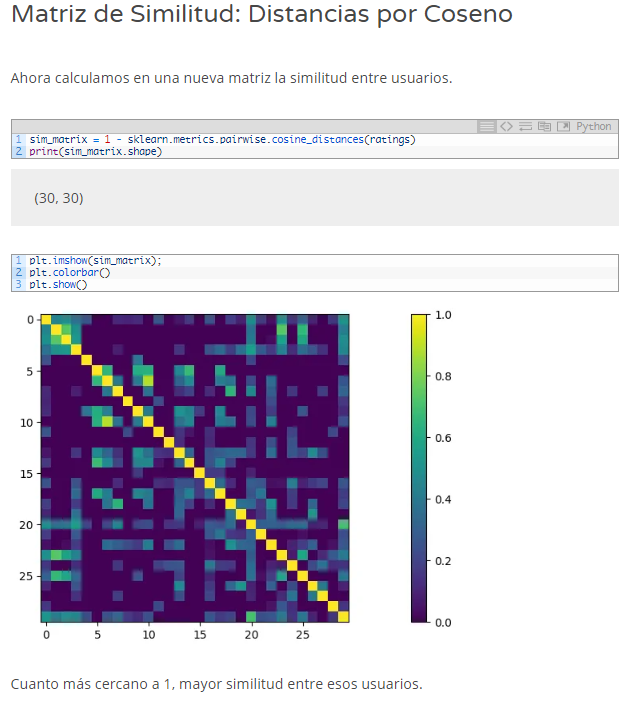
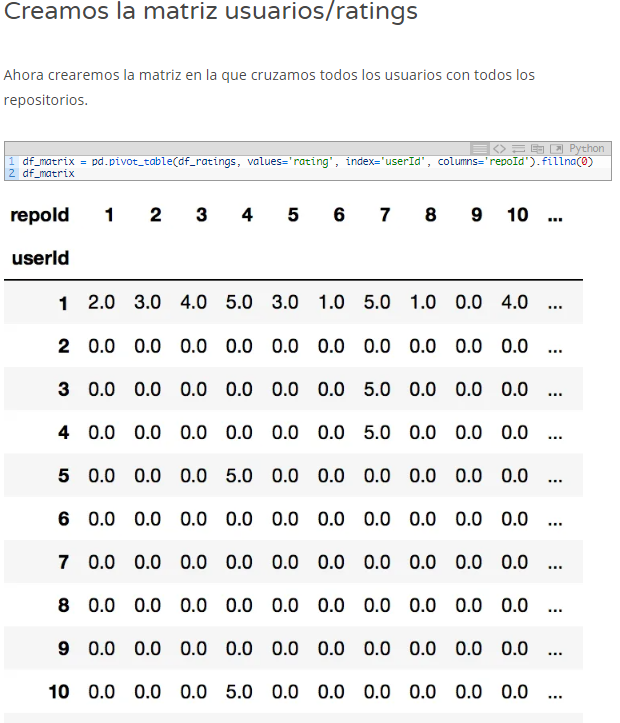
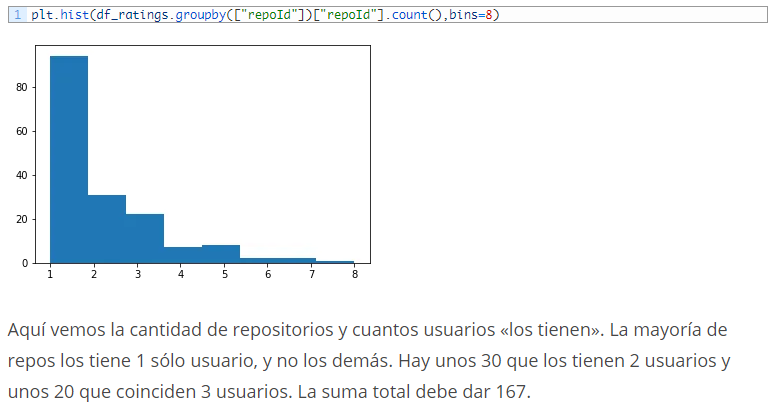
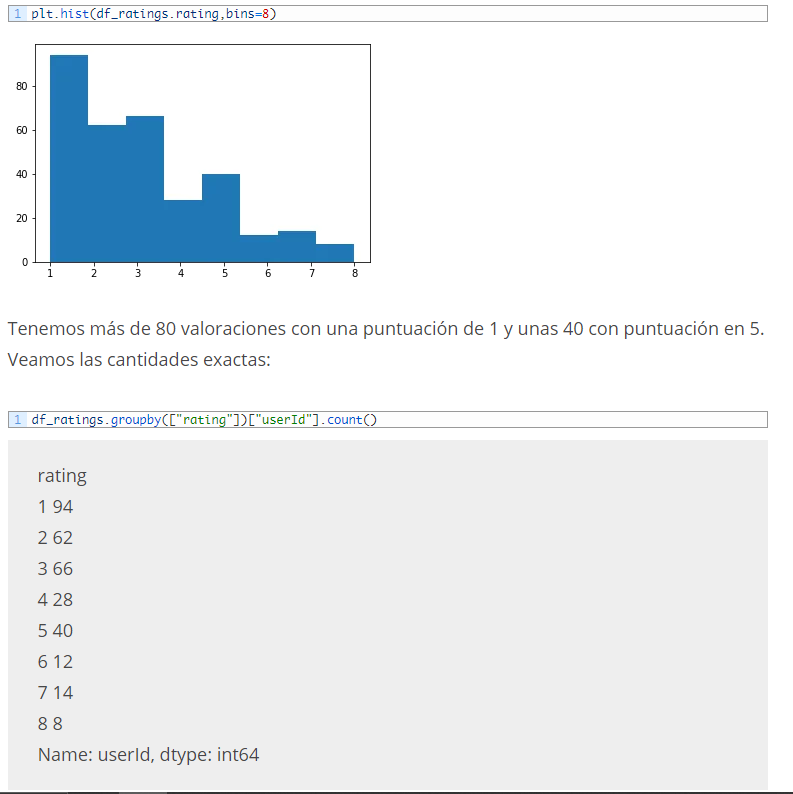
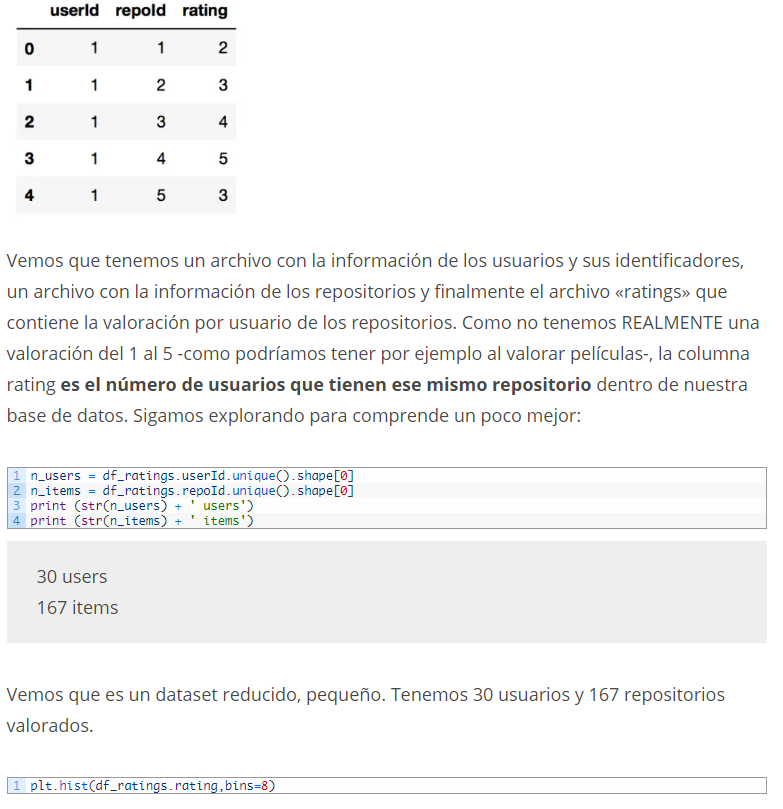
# MARCO TEORICO

Python

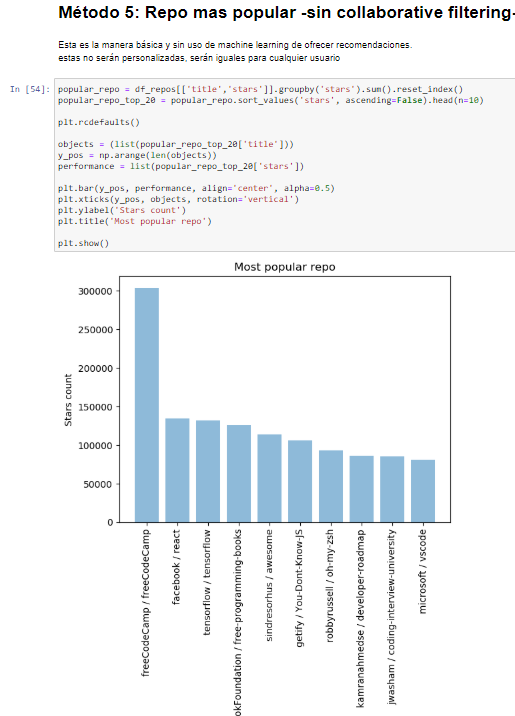
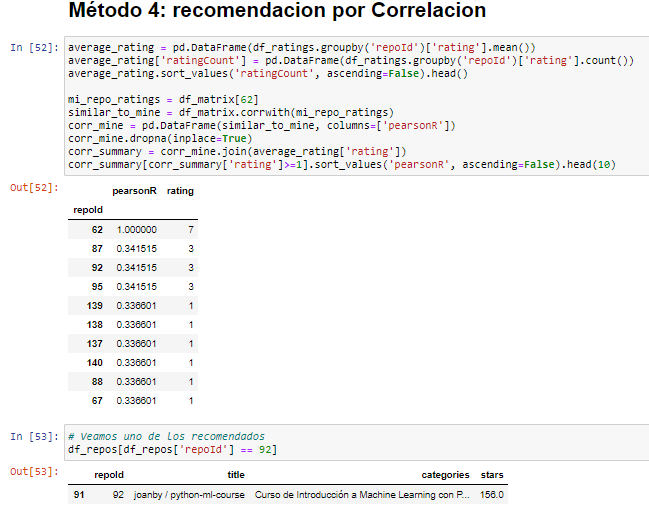
Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. ​ Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

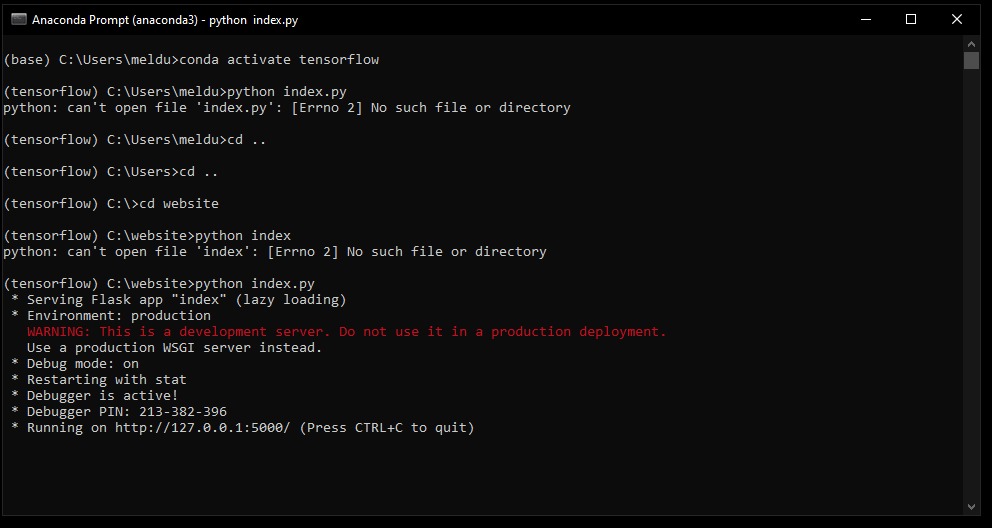
# DESARROLLO

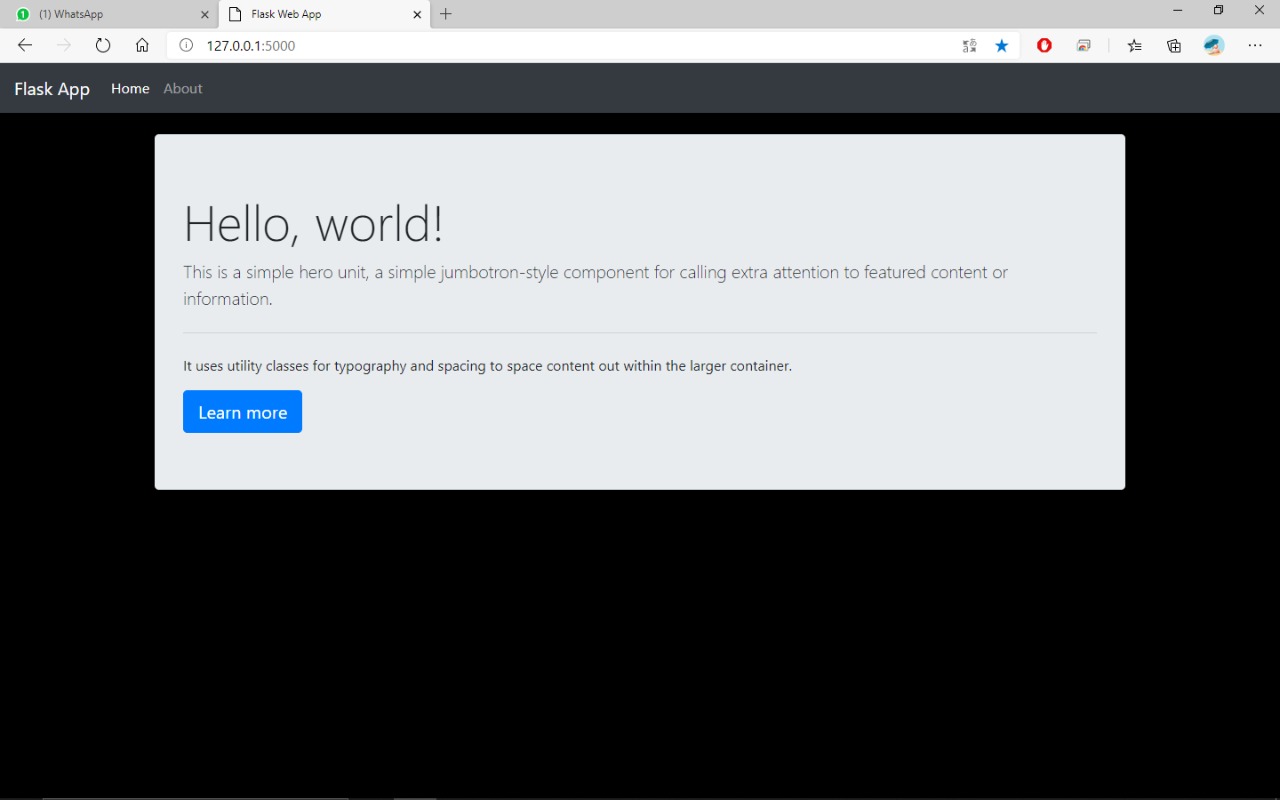




# Hito3







**HITO 4**

import base64

import datetime

import io

import dash

from dash.dependencies import Input, Output, State

import dash\_core\_components as dcc

import dash\_html\_components as html

import dash\_table

import pandas as pd

external\_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(\_\_name\_\_, external\_stylesheets=external\_stylesheets)

app.layout = html.Div([

dcc.Upload(

id='upload-data',

children=html.Div([

'Selecciona o Arrastra tus archivos csv o ',

html.A('Selecciona archivos')

]),

style={

'width': '100%',

'height': '60px',

'lineHeight': '60px',

'borderWidth': '1px',

'borderStyle': 'dashed',

'borderRadius': '5px',

'textAlign': 'center',

'margin': '10px'

},

# Allow multiple files to be uploaded

multiple=True

),

html.Div(id='output-data-upload'),

])

def parse\_contents(contents, filename, date):

content\_type, content\_string = contents.split(',')

decoded = base64.b64decode(content\_string)

try:

if 'csv' in filename:

# Assume that the user uploaded a CSV file

df = pd.read\_csv(

io.StringIO(decoded.decode('utf-8')))

elif 'xls' in filename:

# Assume that the user uploaded an excel file

df = pd.read\_excel(io.BytesIO(decoded))

except Exception as e:

print(e)

return html.Div([

'There was an error processing this file.'

])

return html.Div([

html.H5(filename),

html.H6(datetime.datetime.fromtimestamp(date)),

dash\_table.DataTable(

data=df.to\_dict('records'),

columns=[{'name': i, 'id': i} for i in df.columns]

),

html.Hr(), # horizontal line

# For debugging, display the raw contents provided by the web browser

html.Div('Raw Content'),

html.Pre(contents[0:200] + '...', style={

'whiteSpace': 'pre-wrap',

'wordBreak': 'break-all'

})

])

@app.callback(Output('output-data-upload', 'children'),

[Input('upload-data', 'contents')],

[State('upload-data', 'filename'),

State('upload-data', 'last\_modified')])

def update\_output(list\_of\_contents, list\_of\_names, list\_of\_dates):

if list\_of\_contents is not None:

children = [

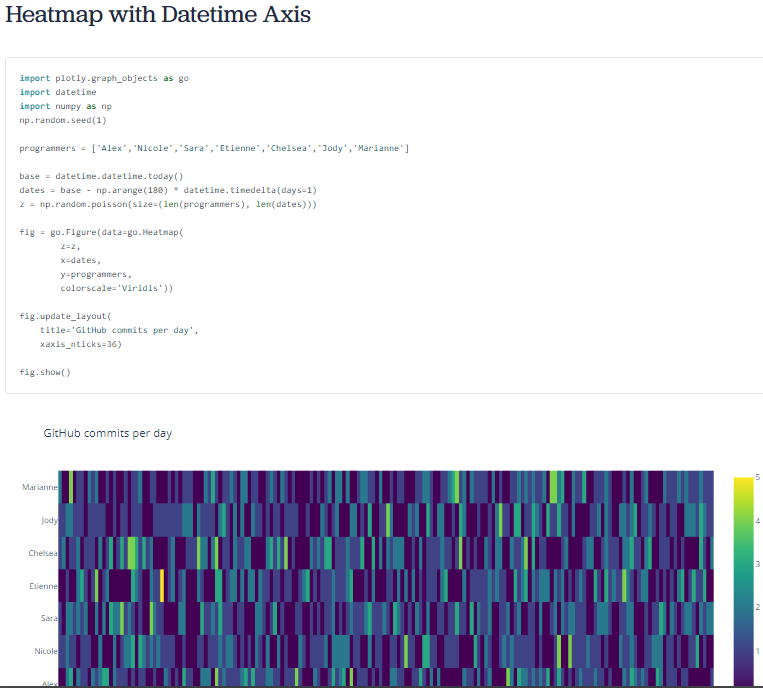
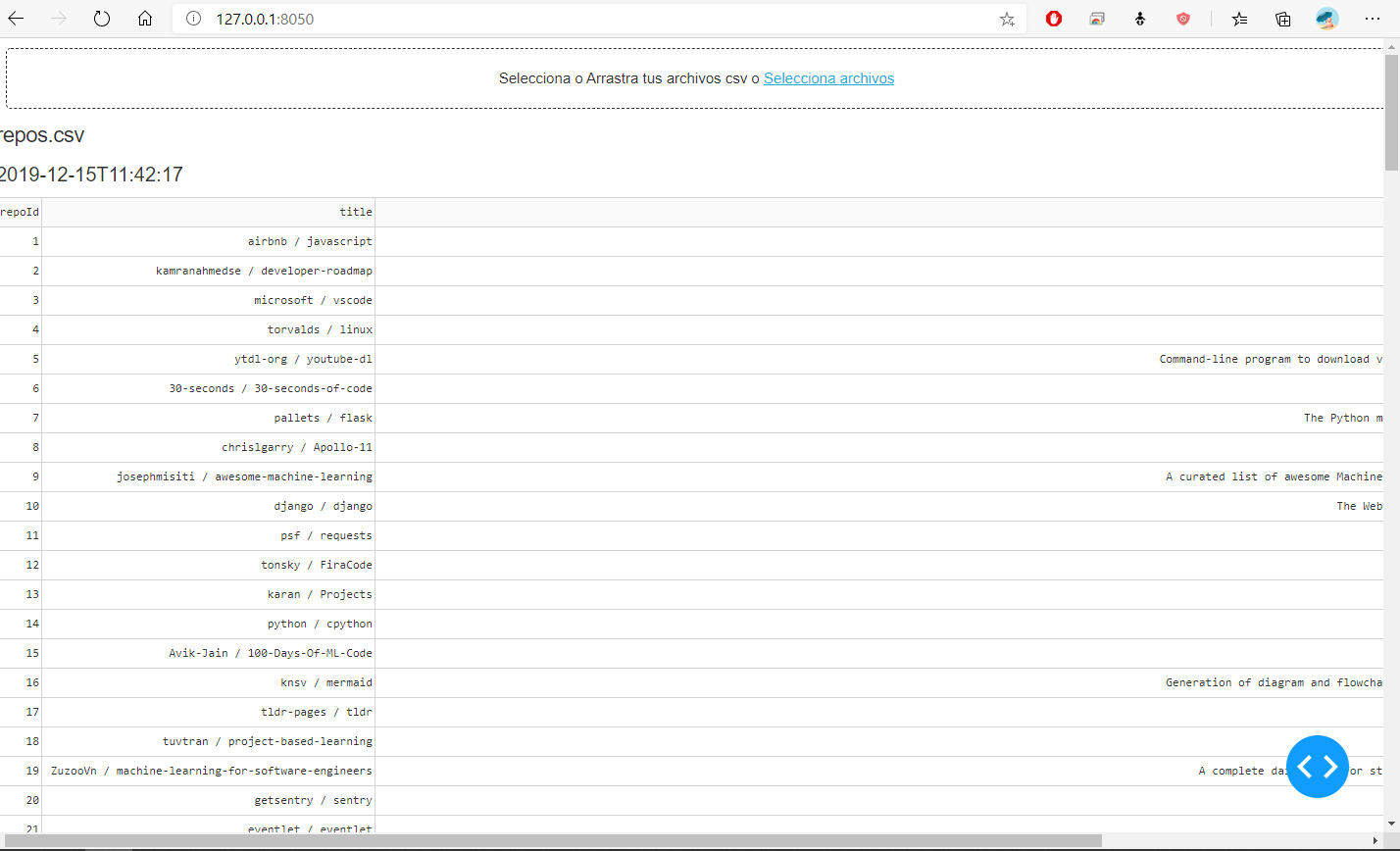
parse\_contents(c, n, d) for c, n, d in

zip(list\_of\_contents, list\_of\_names, list\_of\_dates)]

return children

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True)



# CONCLUSIONES

Vimos que es relativamente sencillo crear un sistema de recomendación en Python y con Machine Learning. Se debe tener volúmenes altos de información. También es central el valor que utilizaremos como rating -siendo una valoración real de cada usuario o un valor artificial que creemos adecuado-. Luego será cuestión de evaluar entre las opciones de motores user-based, ítem-based y seleccionar la que menor error tenga.