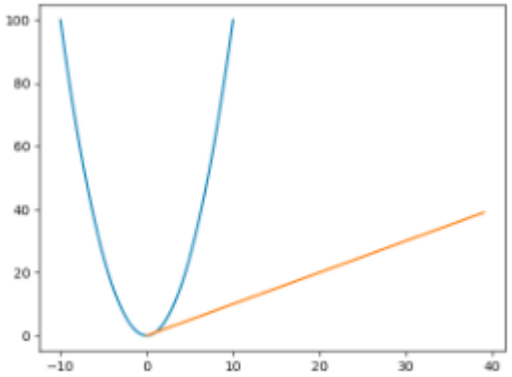
ej1

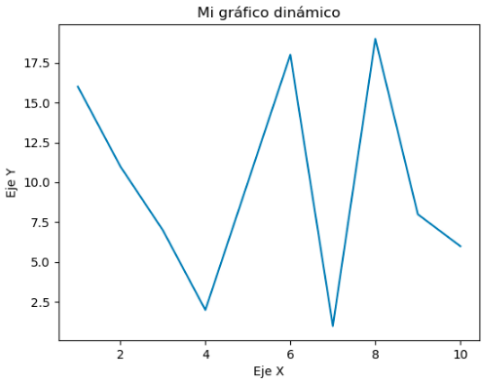


Usando maplotlib se puede grafica

Aquí se grafican una parábola y una recta, pero la idea será graficar una cantidad de puntos correspondientes a valores a los cuales se les aplicará Regresión Lineal

```
from django.shortcuts import render
import matplotlib.pyplot as plt
import io
import urllib, base64
import numpy as np

def home(request):
    # define eje X
    x = np.linspace(-10, 10, 1000)

    # define una parabola
    y = x**2

    # define el grafico y lo realiza
    fig, ax = plt.subplots()
    ax.plot(x, y)

    # crea una recta
    plt.plot(range(40))

    # convierte grafico a imagen
    fig = plt.gcf()
    buf = io.BytesIO()
    fig.savefig(buf,format='png')
    buf.seek(0)

    # codifica la imagen
    string = base64.b64encode(buf.read())
    uri =  urllib.parse.quote(string)

    # la muestra en la pagina html
    return render(request,'home.html',{'data':uri})
```

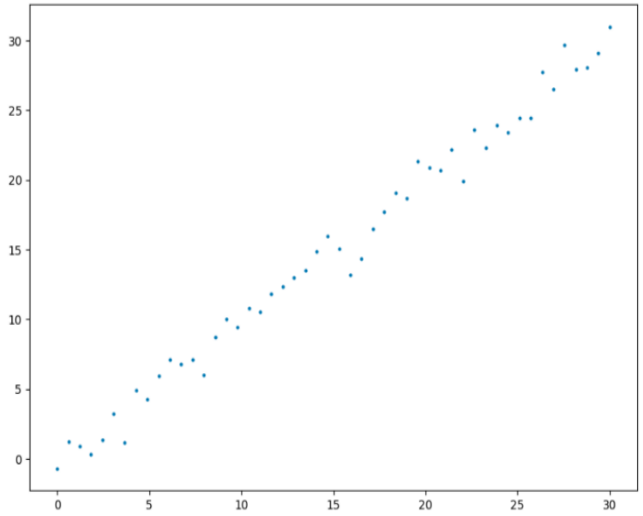
ej2



En este experimento logro un gráfico que se puede manipular con el ratón

```python
from django.http import HttpResponse
from django.shortcuts import render
from matplotlib.backends.backend_agg import FigureCanvasAgg
from random import sample

def home(request):
    return render(request, "core/home.html")

def home(request):
    # datos para representar en el gráfico
    x = range(1,11)
    y = sample(range(20), len(x))
    f = plt.figure() # se crea una figura y se dibuja

    # se crean los ejes
    axes = f.add_axes([0.15, 0.15, 0.75, 0.75]) # [left, bottom, width, height]
    axes.plot(x, y)
    axes.set_xlabel("Eje X")
    axes.set_ylabel("Eje Y")
    axes.set_title("Mi gráfico dinámico")

    # Como enviaremos la imagen en bytes la guardaremos en un buffer
    buf = io.BytesIO()
    canvas = FigureCanvasAgg(f)
    canvas.print_png(buf)

    # Creamos la respuesta enviando los bytes en tipo imagen png
    response = HttpResponse(buf.getvalue(), content_type='image/png')

    # Limpiamos la figura para liberar memoria
    f.clear()

    # Añadimos la cabecera de longitud de fichero para más estabilidad
    response['Content-Length'] = str(len(response.content))

    # Devolvemos la response
    return response
```

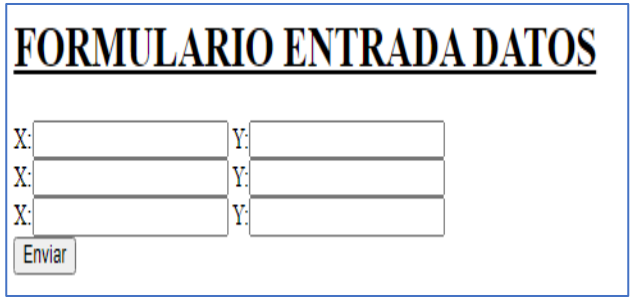
ej3



Aquí se generan puntos aleatorios, pero siguiendo un patrón lineal, para obtener una nube de puntos

```python
from django.shortcuts import render
import matplotlib.pyplot as plt
import random
import io
import urllib, base64
import numpy as np

def home(request):
    n = 50
    for i in range(10):
        x = np.linspace(0, 30, n)

        # se logra un patrón calculando 'y'
        # sumando un valor aleatorio a la 'x'
        y = x + np.random.randn(n)
        plt.figure(figsize=(10,7))
        fig = plt.plot(x,y,'o',markersize=2)

    fig = plt.gcf()
    buf = io.BytesIO()
    fig.savefig(buf,format='png')
    buf.seek(0)
    string = base64.b64encode(buf.read())
    uri =  urllib.parse.quote(string)
    return render(request,'home.html',{'data':uri})
```

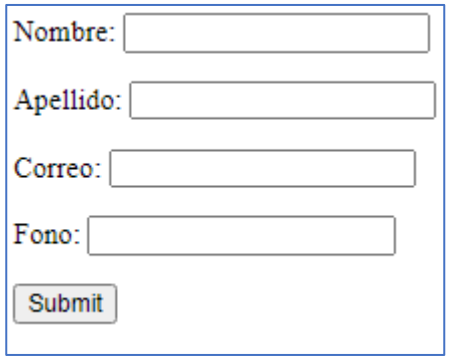
ej4



Experimento para crear un formulario de entrada

```
from django.shortcuts import render
from django.http import  HttpResponse
from .formu import Valorform

def home(request):
    return render(request, 'index.html')

def form(request):
    formulario = Valorform()
    return render(request, 'formu.html', {"form": formulario})
```

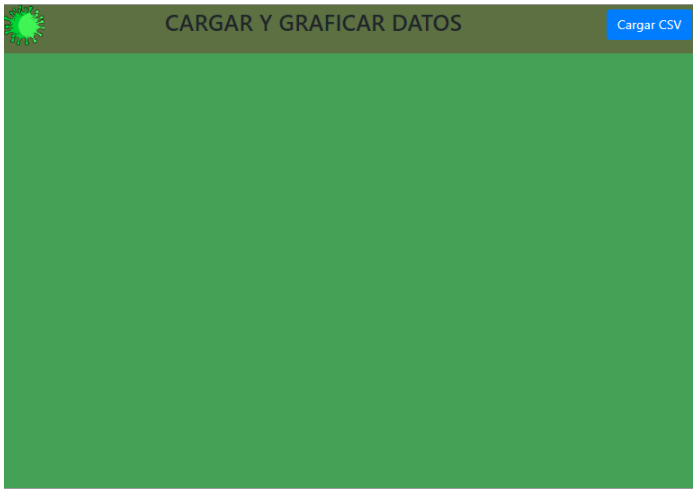
ej5



Aquí un formulario para ingresar datos de entidades

```python
from django.shortcuts import render
from . forms import FormUsuario

def index(request):
    submitbutton= request.POST.get("submit")

    nombre=''
    apellido=''
    correo=''
    fono=''

    form= FormUsuario(request.POST or None)
    if form.is_valid():
        nombre= form.cleaned_data.get("nombre")
        apellido= form.cleaned_data.get("apellido")
        correo= form.cleaned_data.get("correo")
        fono= form.cleaned_data.get("fono")


    context= {'form': form, 'nombre': nombre, 'apellido': apellido,
              'correo': correo, 'fono': fono, 'submitbutton': submitbutton}

    return render(request, 'index.html', context)
```

ej6



Al iniciarla muestra esta página, cuyo objetivo es cargar un CSV con pares de valores a graficar

Luego se modificará esta aplicación para que realice Regresión Lineal con ese conjunto de datos

Al hacer Clic en botón Cargar CSV lee un archivo de tipo CSV que contiene pares de valores x, y. Entonces los grafica