

# Proyecto Corto - Reto STEAM Explorando Datos de Pacientes en una Clínica\_Mariana\_Villalobos

February 15, 2025

## 1 Proyecto Corto: Explorando Datos de Pacientes en una Clínica

### 1.1 Programacion Python Basico | ICAI - UNA

### 1.2 Profesor: Ing. Andrés Mena Abarca

#### 1.2.1 Nombre del estudiante: Mariana Villalobos Vargas

---

¡Bienvenidos al reto STEAM! Hoy serás parte de un equipo de científicos de datos que está desarrollando una aplicación para analizar los datos de los pacientes de una clínica. Este proyecto se integra dentro del enfoque STEAM, donde no solo usaremos habilidades de programación y análisis, sino también creatividad, visualización y reflexión crítica para mejorar la experiencia de salud.

En este reto, utilizaremos **Python** para analizar datos clínicos reales y responder preguntas importantes sobre la salud de los pacientes. También diseñarás soluciones creativas y colaborarás con tus compañeros para mejorar la aplicación.

#### 1.2.2 Contexto:

El director de la clínica te ha encargado que construyas un sistema que pueda analizar la información de los pacientes, entender las enfermedades tratadas, los medicamentos recetados y comparar pacientes.

#### 1.2.3 Objetivos del Reto:

1. Aplicar estructuras básicas de Python como listas y ciclos.
2. Leer datos desde un archivo JSON y almacenarlos en listas.
3. Realizar análisis de datos sobre enfermedades y medicamentos.
4. Visualizar y comunicar los resultados de manera creativa.

#### 1.2.4 Fase Creativa STEAM:

- **Ciencia y Tecnología:** Analizarás los datos clínicos.
- **Ingeniería:** Implementarás las soluciones utilizando estructuras de Python.
- **Arte y Matemáticas:** Visualizarás los resultados de manera atractiva y clara.

¡Comencemos!

### 1.3 Cargar y Leer los Datos de Pacientes

```
[3]: import json

with open('clinica_intermedio.json', encoding='utf-8') as archivo:
    datos = json.load(archivo)
```

### 1.4 Desafío 1: Análisis Científico - Reporte de Enfermedades Tratadas

Tu primer reto es analizar los datos de los pacientes para generar un reporte de las enfermedades tratadas en la clínica. Utiliza listas y ciclos para contar cuántos pacientes han sido tratados por cada enfermedad.

#### 1.4.1 Instrucciones:

1. Crea una lista vacía donde almacenarás las enfermedades.
2. Recorre los datos de los pacientes y extrae la enfermedad de cada paciente.
3. Cuenta cuántos pacientes han sido tratados por cada enfermedad.
4. Visualiza los resultados de manera clara utilizando la técnica que elijas (gráficos o tablas).

#### 1.4.2 Preguntas:

- ¿Qué enfermedades son las más tratadas en la clínica?
- ¿Hay alguna enfermedad que te sorprenda por su prevalencia?

#### 1.4.3 Paso a Paso para el Estudiante

##### Paso 1: Crear las Listas

##### 1.4.4

1. **Objetivo:** Necesitamos dos listas para almacenar:
  - Los nombres de las enfermedades.
  - La cantidad de veces que cada enfermedad aparece.
2. **Instrucción:**
  - Crea una lista vacía llamada `lista_enfermedades`.
  - Crea otra lista vacía llamada `lista_cantidades`.

```
[4]: # Lista para almacenar las enfermedades y sus cantidades
lista_enfermedades = []
lista_cantidades = []
```

##### Paso 2: Recorrer los Datos de los Pacientes

##### 1.4.5

1. **Objetivo:** Vamos a analizar cada paciente para extraer su enfermedad.
2. **Instrucción:**

- Usa un bucle `for` para recorrer cada `paciente` en `datos_pacientes` (ya está definida en el código como una lista de listas).
- Dentro del bucle, asigna la enfermedad del paciente (índice 4 de cada lista) a una variable llamada `enfermedad`.

```
[5]: # Extraemos la enfermedad del paciente
enfermedad = []
for paciente in datos:
    enfermedad.append(paciente[4])
#Puedes imprimir el for para ver los datos que se procesan
print (enfermedad)
```

```
['gripe', 'gastritis', 'tos', 'migraña', 'dolor', 'gripe', 'gastritis',
'diabetes', 'asma', 'gripe', 'hipotiroidismo', 'alergias', 'migraña', 'presión
alta', 'dolor de espalda', 'ansiedad', 'diabetes', 'asma', 'alergias',
'gastritis', 'presión alta', 'tos', 'dolor de espalda', 'ansiedad', 'diabetes',
'asma', 'alergias', 'hipotiroidismo', 'gripe', 'gastritis', 'migraña', 'presión
alta', 'asma', 'diabetes', 'dolor', 'gastritis', 'presión alta', 'asma',
'gripe']
```

### Paso 3: Verificar si la Enfermedad ya Está en la Lista

1. **Objetivo:** Verificar si la enfermedad ya fue registrada.
2. **Instrucción:**
  - Usa un `if` para verificar si la `enfermedad` ya está en `lista_enfermedades`.
  - Si **no** está en `lista_enfermedades`, agrega la enfermedad a `lista_enfermedades` y añade 1 a `lista_cantidades` para contarla por primera vez.
  - Si **sí** está en `lista_enfermedades`, encuentra la posición (`índice`) de la enfermedad en `lista_enfermedades` y aumenta en 1 el valor en `lista_cantidades` en la misma posición.

```
[6]: # Pista: Usa `in` para verificar si la enfermedad ya está en lista_enfermedades.
# Pista: Si la enfermedad no está, agrégala a lista_enfermedades y empieza el
↪ conteo en lista_cantidades.
# Pista: Si la enfermedad ya está, usa `index` para encontrar su posición en
↪ lista_enfermedades y suma 1 en lista_cantidades en esa posición.
lista_enfermedades = []
lista_cantidades = []

for dato in enfermedad:
    if dato not in lista_enfermedades:
        lista_enfermedades.append(dato)
        lista_cantidades.append([dato,1])
    else:
        indice = lista_enfermedades.index(dato)
        lista_cantidades[indice][1] += 1
```

```
#Puedes imprimir el for para ver los datos que se procesan
print (lista_enfermedades)
print (lista_cantidades)
```

```
['gripe', 'gastritis', 'tos', 'migraña', 'dolor', 'diabetes', 'asma',
'hipotiroidismo', 'alergias', 'presión alta', 'dolor de espalda', 'ansiedad']
[['gripe', 5], ['gastritis', 5], ['tos', 2], ['migraña', 3], ['dolor', 2],
['diabetes', 4], ['asma', 5], ['hipotiroidismo', 2], ['alergias', 3], ['presión
alta', 4], ['dolor de espalda', 2], ['ansiedad', 2]]
```

#### Paso 4: Imprimir el Reporte Final

1. **Objetivo:** Mostrar el número de pacientes con cada enfermedad.

2. **Instrucción:**

- Usa un bucle for para recorrer la lista `lista_enfermedades`.
- En cada iteración, imprime la enfermedad y el número de pacientes (que está en `lista_cantidades` en la misma posición).

```
[26]: print("Reporte de enfermedades tratadas:")
      for dato in lista_cantidades:
          print (f'{dato[1]} pacientes con {dato[0]}.'
```

```
Reporte de enfermedades tratadas:
5 pacientes con gripe.
5 pacientes con gastritis.
2 pacientes con tos.
3 pacientes con migraña.
2 pacientes con dolor.
4 pacientes con diabetes.
5 pacientes con asma.
2 pacientes con hipotiroidismo.
3 pacientes con alergias.
4 pacientes con presión alta.
2 pacientes con dolor de espalda.
2 pacientes con ansiedad.
```

#### 1.4.6 Resumen Completo del Código

En ésta celda resume el código completo que debería tener al final para la solución del ejercicio:

```
[25]: #Esta parte puedes colocar todos los bloques de las celdas anteriores
import json
with open('clinica_intermedio.json', encoding='utf-8') as archivo:
    datos = json.load(archivo)

enfermedad = []
for paciente in datos:
    enfermedad.append(paciente[4])
```

```

print (enfermedad)
lista_enfermedades = []
lista_cantidades = []

for dato in enfermedad:
    if dato not in lista_enfermedades:
        lista_enfermedades.append(dato)
        lista_cantidades.append([dato,1])
    else:
        indice = lista_enfermedades.index(dato)
        lista_cantidades[indice][1] += 1

print (lista_enfermedades)
print (lista_cantidades)
print("\nReporte de enfermedades tratadas:\n")
for dato in lista_cantidades:
    print (f'{dato[1]} pacientes con {dato[0]}'.)

```

```

['gripe', 'gastritis', 'tos', 'migraña', 'dolor', 'gripe', 'gastritis',
'diabetes', 'asma', 'gripe', 'hipotiroidismo', 'alergias', 'migraña', 'presión
alta', 'dolor de espalda', 'ansiedad', 'diabetes', 'asma', 'alergias',
'gastritis', 'presión alta', 'tos', 'dolor de espalda', 'ansiedad', 'diabetes',
'asma', 'alergias', 'hipotiroidismo', 'gripe', 'gastritis', 'migraña', 'presión
alta', 'asma', 'diabetes', 'dolor', 'gastritis', 'presión alta', 'asma',
'gripe']
['gripe', 'gastritis', 'tos', 'migraña', 'dolor', 'diabetes', 'asma',
'hipotiroidismo', 'alergias', 'presión alta', 'dolor de espalda', 'ansiedad']
[['gripe', 5], ['gastritis', 5], ['tos', 2], ['migraña', 3], ['dolor', 2],
['diabetes', 4], ['asma', 5], ['hipotiroidismo', 2], ['alergias', 3], ['presión
alta', 4], ['dolor de espalda', 2], ['ansiedad', 2]]

```

Reporte de enfermedades tratadas:

```

5 pacientes con gripe.
5 pacientes con gastritis.
2 pacientes con tos.
3 pacientes con migraña.
2 pacientes con dolor.
4 pacientes con diabetes.
5 pacientes con asma.
2 pacientes con hipotiroidismo.
3 pacientes con alergias.
4 pacientes con presión alta.
2 pacientes con dolor de espalda.
2 pacientes con ansiedad.

```

## 1.5 Desafío 2: Exploración Tecnológica - Reporte de Medicamentos Recetados

Ahora que ya tienes el reporte de las enfermedades tratadas, es momento de explorar qué medicamentos se están recetando en la clínica. Tu tarea es generar un reporte similar al anterior, pero esta vez debes contar cuántos pacientes han recibido cada medicamento.

### 1.5.1 Instrucciones:

1. Almacena los medicamentos recetados en una lista.
2. Recorre los datos y cuenta cuántos pacientes han recibido cada medicamento.
3. Visualiza los resultados de manera creativa.

### 1.5.2 Preguntas:

- ¿Qué medicamentos son los más recetados?
- ¿Cómo podrías optimizar el uso de medicamentos en la clínica?

Resultados esperados:

Reporte de medicamentos recetados:

acetaminofen: 9 pacientes

antiácido: 2 pacientes

jarabe: 2 pacientes

insulina: 3 pacientes

salbutamol: 4 pacientes

levotiroxina: 2 pacientes

.....

```
[ ]: # Listas para almacenar los medicamentos y sus frecuencias
from tabulate import tabulate
medicamentos = []
frecuencias = []

for paciente in datos:
    medicamentos.append(paciente[5])

lista_medicamentos = []
lista_frecuencias = []

for dato in medicamentos:
    if dato not in lista_medicamentos:
        lista_medicamentos.append(dato)
        lista_frecuencias.append([dato,1])
    else:
        indice = lista_medicamentos.index(dato)
        lista_frecuencias[indice][1] += 1

lista_frecuencias.sort(key=lambda x: x[1], reverse=True) # Ordenar la lista por
↳ la cantidad de pacientes de mayor a menor
```

```
print("\nReporte de medicamentos recetados:\n")
print(tabulate(lista_frecuencias, headers=["Medicamento", "Cantidad de_
↪pacientes"], tablefmt="grid"))
#Se utiliza la librería tabulate con el formato grid para presentar los datos_
↪de manera clara.
```

Reporte de medicamentos recetados:

Medicamento	Cantidad de pacientes
acetaminofen	9
salbutamol	4
insulina	3
antiácido	2
jarabe	2
levotiroxina	2
loratadina	2
amlodipino	2
ibuprofeno	2
omeprazol	2
clonazepam	1
cetirizina	1
losartán	1
paracetamol	1
diazepam	1
metformina	1
fluticasona	1
pantoprazol	1

enalapril		1	
+-----+			

```
[ ]: #Preguntas:
# ¿Qué medicamentos son los más recetados?

lista_frecuencias.sort(key=lambda x: x[1], reverse=True)
print(f'El medicamento de mayor uso es: {lista_frecuencias[0][0]}')
```

El medicamento de mayor uso es: acetaminofen.

### 1.5.3 ¿Cómo podrías optimizar el uso de medicamentos en la clínica?

- Llevar una lista de medicamentos recetados no retirados para concientizar sobre el desperdicio.
- Comprar en mayor cantidad los medicamentos más utilizados.
- Evaluar el patrón de prescripción para evitar el sobreuso.

## 1.6 Desafío 3: Ingeniería - Comparar Pacientes por Identificación

Este desafío te permitirá comparar dos pacientes de la clínica en base a sus identificaciones. Tu objetivo es averiguar:

1. ¿Qué enfermedades tienen en común?
2. ¿Qué medicamentos toman en común?

### 1.6.1 Instrucciones:

1. Solicita dos identificaciones de pacientes al usuario.
2. Extrae las enfermedades y medicamentos de esos dos pacientes.
3. Compara las enfermedades y medicamentos y muestra los resultados.

### 1.6.2 Preguntas:

- ¿Qué enfermedades y medicamentos son comunes en los pacientes comparados?
- ¿Qué información adicional podrías incluir para mejorar esta comparación?

```
[62]: # Pedimos las identificaciones de los dos pacientes
pacientes_comparados= []

while len(pacientes_comparados) < 2:
    pacientes_comparados.append(input('Favor ingrese número de identificación:
    ↵'))

datos_pacientes=[]

for id in pacientes_comparados:
    for paciente in datos:
        if id == paciente[0]:
            datos_pacientes.append([paciente[0],paciente[4], paciente[5]])
```



```

print(datos_pacientes)

enfermedades = [paciente[1] for paciente in datos_pacientes]
if enfermedades[0] == enfermedades[1]:
    print(f"\nEnfermedades comunes: {enfermedades[0]}.")
else:
    print("\nNo hay enfermedades comunes.")

medicamentos = [paciente[2] for paciente in datos_pacientes]
if medicamentos[0] == medicamentos[1]:
    print(f"\nMedicamentos comunes: {medicamentos[0]}.")
else:
    print("\nNo hay medicamentos comunes.")

#603240305 y 116130052

```

```

[['603240305', 'gastritis', 'antiácido'], ['116130052', 'gastritis',
'antiácido']]

```

Enfermedades comunes: gastritis.

Medicamentos comunes: antiácido.

### 1.6.3 Información adicional que se puede incluir para mejorar la comparación:

- Identificación del paciente
- Nombre del paciente, pero la información no viene en la lista

```

[63]: pacientes_comparados= []

while len(pacientes_comparados) < 2:
    pacientes_comparados.append(input('Favor ingrese número de identificación:
↵'))

datos_pacientes=[]

for id in pacientes_comparados:
    for paciente in datos:
        if id == paciente[0]:
            datos_pacientes.append([paciente[0],paciente[4], paciente[5]])

print(datos_pacientes)

enfermedades = [paciente[1] for paciente in datos_pacientes]
if enfermedades[0] == enfermedades[1]:
    print(f"\nLos pacientes con ID {datos_pacientes[0][0]} y
↵{datos_pacientes[1][0]} tienen como enfermedades comunes: {enfermedades[0]}.
↵")

```

```

else:
    print("\nNo hay enfermedades comunes.")

medicamentos = [paciente[2] for paciente in datos_pacientes]
if medicamentos[0] == medicamentos[1]:
    print(f"\nLos pacientes con ID {datos_pacientes[0][0]} y {datos_pacientes[1][0]} tienen como medicamentos comunes: {medicamentos[0]}.")
else:
    print("\nNo hay medicamentos comunes entre los pacientes.")

#603240305 y 116130052

```

```
[['603240305', 'gastritis', 'antiácido'], ['116130052', 'gastritis', 'antiácido']]
```

Los pacientes con ID 603240305 y 116130052 tienen como enfermedades comunes: gastritis.

Los pacientes con ID 603240305 y 116130052 tienen como medicamentos comunes: antiácido.

## 1.7 Desafío Final: Arte y Matemáticas - Visualización Creativa y Reflexión

Has llegado al último desafío, ¡felicidades! Ahora es momento de presentar tus resultados de una forma creativa. Tu tarea será utilizar **gráficos**, **tablas** o incluso **arte digital** para mostrar tus descubrimientos sobre los datos de los pacientes de la clínica.

### 1.7.1 Preguntas:

- ¿Qué patrones encontraste en los datos?
- ¿Cómo podrías usar estos análisis para mejorar la atención de los pacientes?

```

[80]: import matplotlib.pyplot as plt
      from collections import Counter

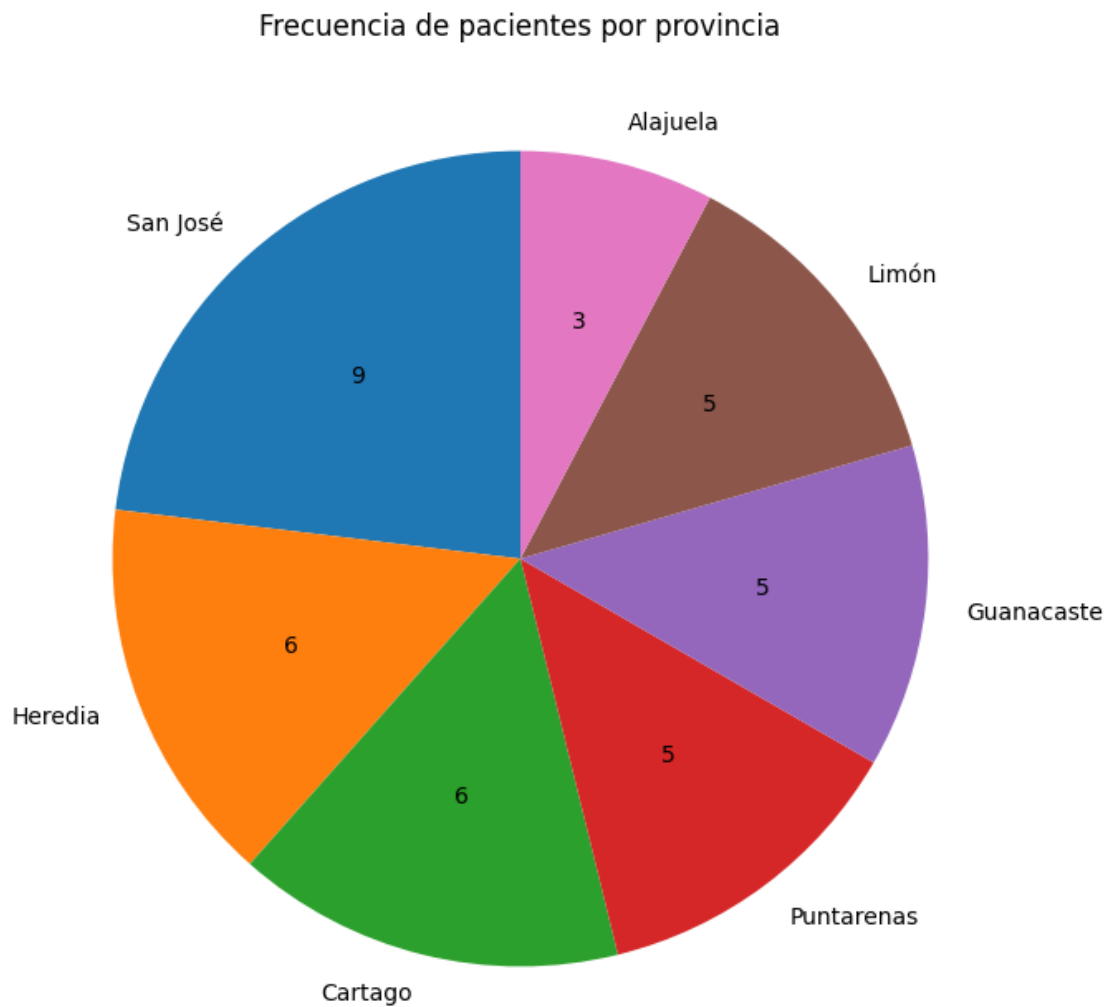
provincias = [paciente[2] for paciente in datos]
frecuencia_prov = Counter(provincias)

provincias_ordenadas = dict(sorted(frecuencia_prov.items(), key=lambda x: x[1],
    ↪reverse=True)) #Ordenar canton por frecuencia.

# Graficar
plt.figure(figsize=(8,8))
plt.pie(provincias_ordenadas.values(), labels=provincias_ordenadas.keys(),
    ↪autopct=lambda p: f'{int(p * sum(provincias_ordenadas.values()) / 100)}%',
    ↪startangle=90)

```

```
plt.title('Frecuencia de pacientes por provincia')
plt.show()
```



### 1.7.2 Patrones de datos, provincias:

- Los pacientes provienen principalmente de San José, Heredia y Cartago.
- De Alajuela, a pesar de ser parte del GAM, proviene la menor cantidad de pacientes.

```
[ ]: import matplotlib.pyplot as plt
from collections import Counter

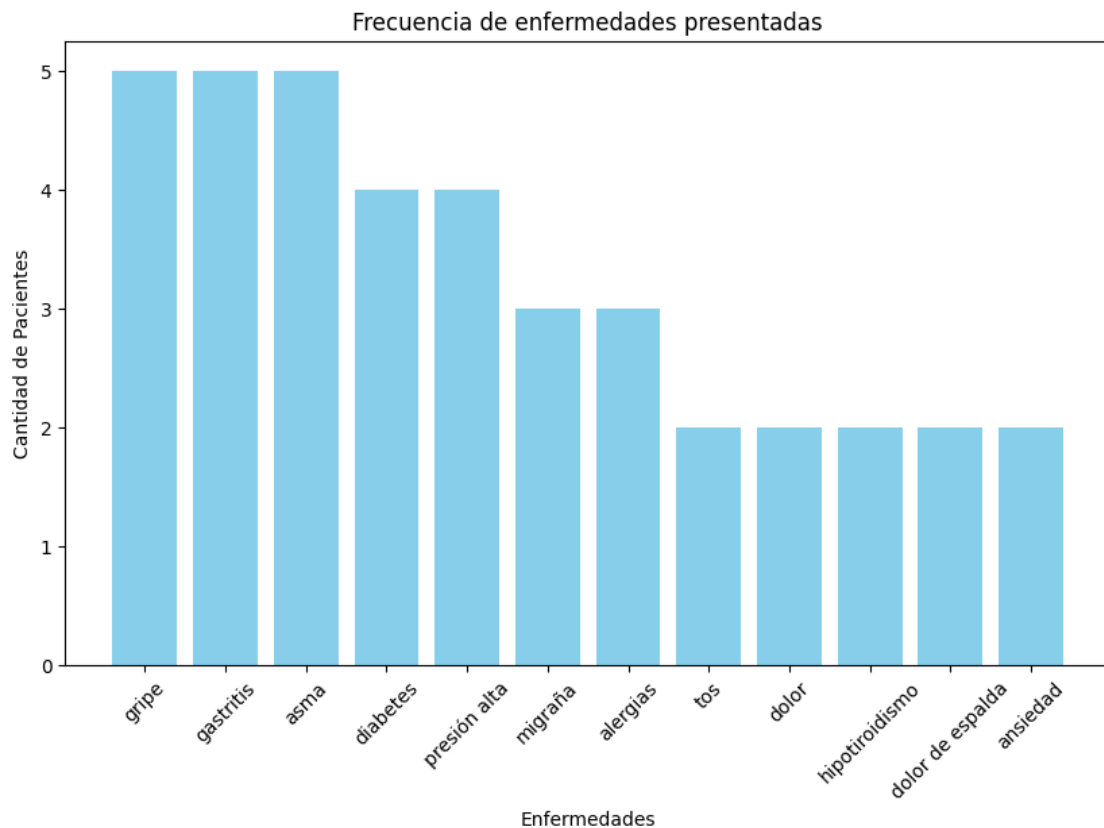
enfermedades = [paciente[4] for paciente in datos]
enfermedades_contadas = Counter(enfermedades)
```

```

enfermedades_ordenadas = dict(sorted(enfermedades_contadas.items(), key=lambda
    ↪x: x[1], reverse=True)) #Ordenar enfermedad por frecuencia.

# Graficar
plt.figure(figsize=(10,6))
plt.bar(enfermedades_ordenadas.keys(), enfermedades_ordenadas.values(),
    ↪color='skyblue')
plt.xlabel('Enfermedades')
plt.ylabel('Cantidad de Pacientes')
plt.title('Frecuencia de enfermedades presentadas')
plt.xticks(rotation=45)
plt.show()

```



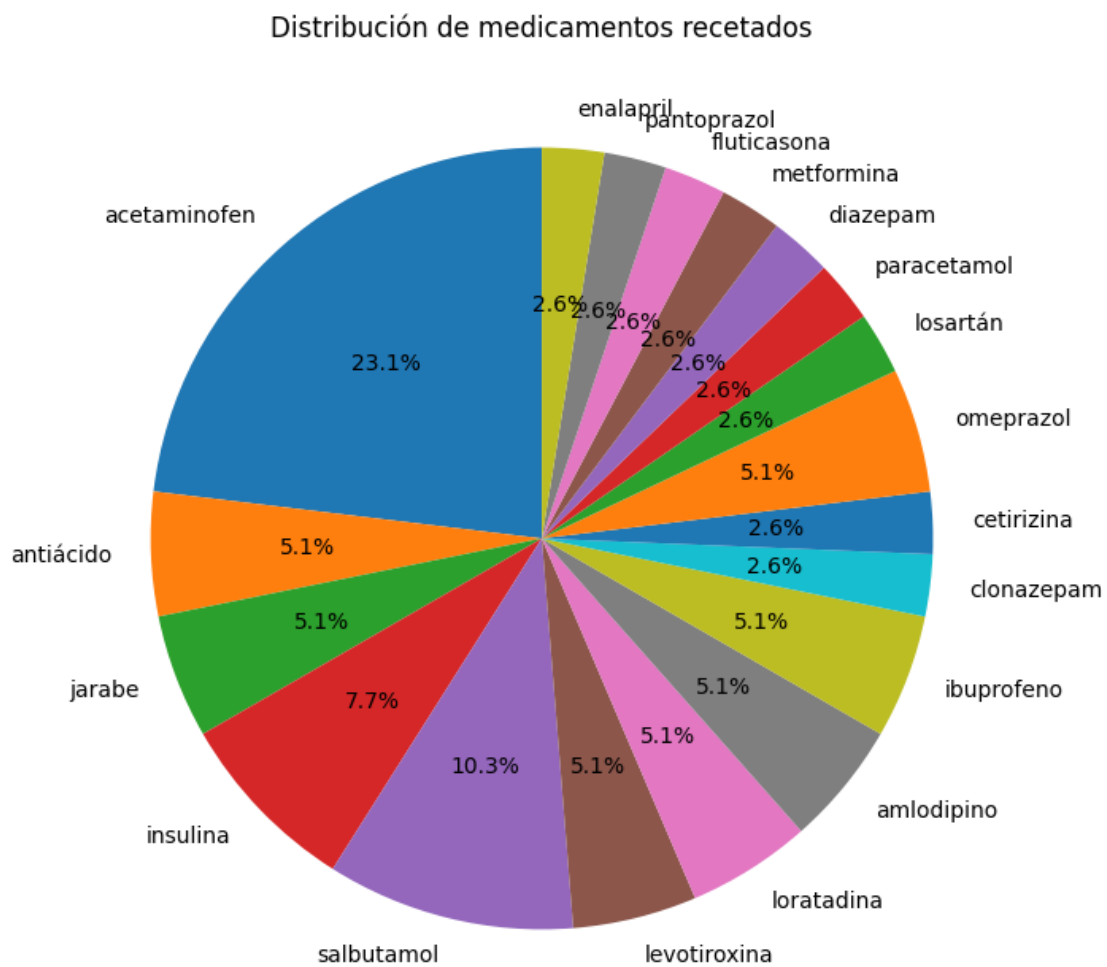
### 1.7.3 Patrones de datos, enfermedades:

- Las enfermedades de mayor frecuencia son la gripe, gastritis y asma.
- De forma poco frecuente se presentan enfermedades como ansiedad y dolor de espalda.
- Diabetes y presión alta son enfermedades de frecuencia intermedia.
- Es necesario valorar las edades de los pacientes que consultan para poder entender el comportamiento de estos datos.

```
[81]: medicamentos_rec = [paciente[5] for paciente in datos]
medicamentos_contados = Counter(medicamentos_rec)

# Contamos las frecuencias de los medicamentos
medicamentos_contados = Counter(medicamentos_rec)

# Gráfico de pastel
plt.figure(figsize=(8,8))
plt.pie(medicamentos_contados.values(), labels=medicamentos_contados.keys(),
        autopct='%1.1f%%', startangle=90)
plt.title('Distribución de medicamentos recetados')
plt.show()
```



#### **1.7.4 Patrones de datos, medicamentos recetados:**

- El medicamento de mayor uso es la acetaminofén. Si se compara con la enfermedad de mayor frecuencia sí tiene sentido, ya que es gripe y para abordar esta enfermedad, al ser viral, normalmente lo único que se realiza es tratamiento de síntomas.
- El segundo medicamento más recetado es antiácido, que también correlaciona con la segunda enfermedad en frecuencia, gastritis.
- Los datos captados solamente permiten una “enfermedad” y un medicamento, pero sabemos que esto no ocurre de esta manera, por lo que es necesaria una base de datos que contemple todas las enfermedades reportadas por los pacientes, así como todos los medicamentos recetados para tener un mejor panorama y poder realizar comparaciones de manera correcta.

#### **1.7.5 ¿Cómo podría usar estos análisis para mejorar la atención de los pacientes?**

- Se puede tomar en cuenta de dónde proviene el paciente con el fin de recetar los medicamentos necesarios oportunamente si es de sitios alejados, de manera que acuda en la menor medida posible al centro de salud, haciendo más eficiente la atención.
- De acuerdo con la frecuencia encontrada en las enfermedades, se puede utilizar en el manejo de inventarios para contar con los medicamentos más recetados siempre en stock y evitar un desabastecimiento eventual.
- Se puede hacer un estudio de la frecuencia de las enfermedades para analizar posibles razones y realizar campañas de salud dirigidas a estas morbilidades, de manera que si es necesaria una intervención sanitaria, se realice oportunamente, tomando en cuenta incluso ubicación geográfica.