

# DISASTER BOT

"Tu asistente catastrófico inteligente"



# Equipo



**Nicolás  
Rodríguez**  
Data Scientist  
@ ACHS



**Fernando  
Duarte**  
Software Engineer  
@ Banca.Me



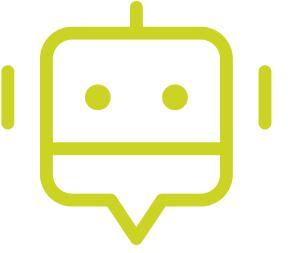
**Max Guzmán**  
Data Engineer  
@ Foris.ai



**Fabián  
Díaz**  
Cloud Solution  
Architect Intern  
@ Microsoft

# Dolor

No estamos bien preparados para los desastres naturales

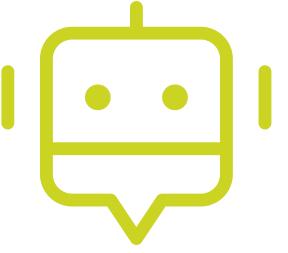


**5.707 personas afectadas**

- Erupción Volcán Chaitén, 2008

# Dolor

No estamos bien preparados para los desastres naturales



Chaitén



5 horas

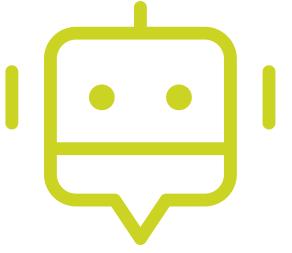
---

Hospital de Ancud



# Dolor

No estamos bien preparados para los desastres naturales



**6 personas fallecidas**



**32 personas heridas**

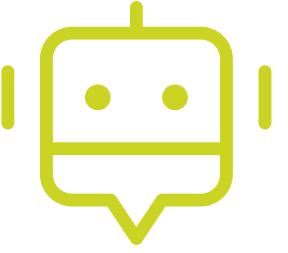


**86 casas destruidas**

- Terremoto de Cobquecura, 2010

# Dolor

No estamos bien preparados para los desastres naturales



Cobquecura



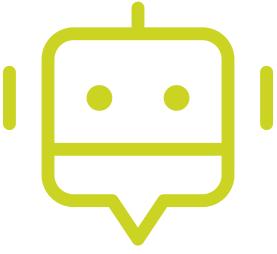
2 horas



Hospital Clínico Regional



# Dolor

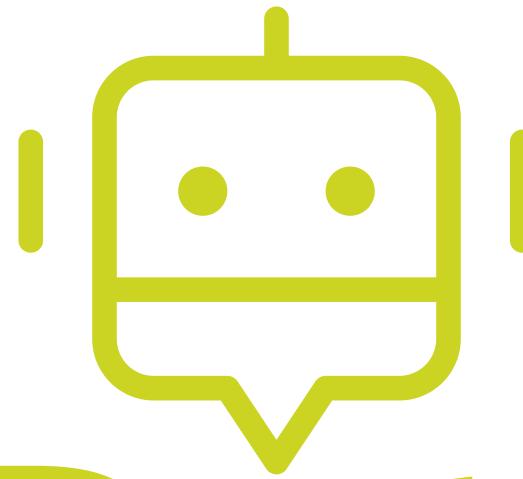


No estamos bien preparados, pero podríamos estarlo

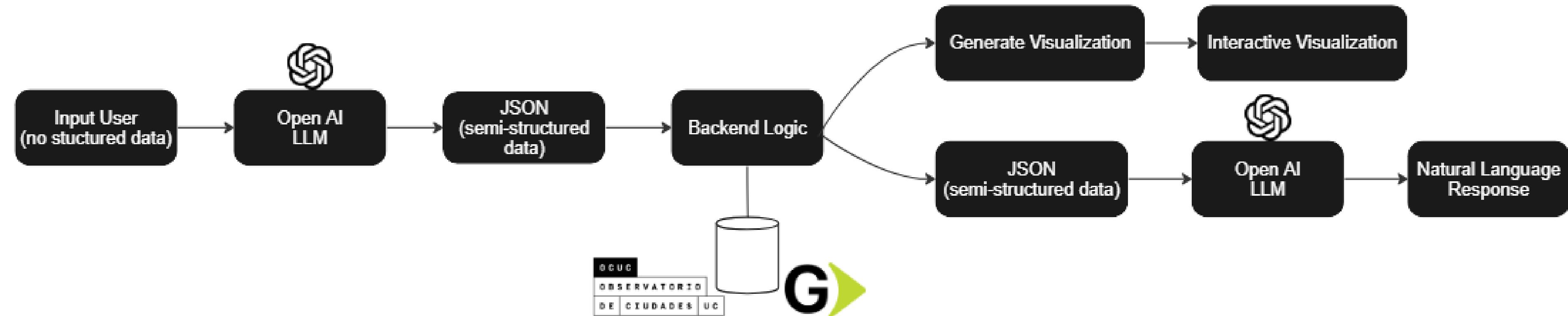
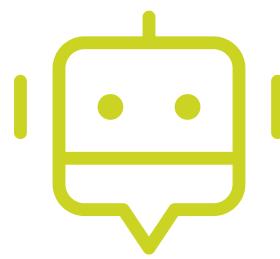
¿Cómo podemos reducir el  
daño?

# Disaster Bot

---



# Arquitectura





# Oportunidad

Estimamos que si logramos reducir un 10% los daños en salud, podríamos ahorrar:

**1M-2M USD**

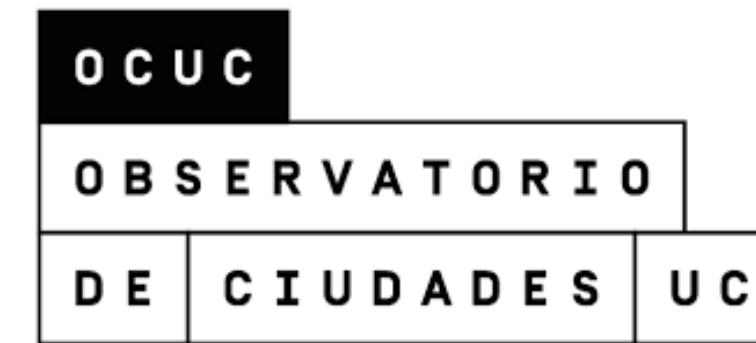
Gracias a la reducción de gasto público en salud

**10-50 Vidas**

Debido al aumento de la atención de urgencia temprana



# Fuentes





# Escalabilidad

The image displays two side-by-side screenshots of a mobile messaging application interface, likely WhatsApp, illustrating the scalability of a disaster response bot named "Disaster Bot".

**Screenshot 1 (Left):**

- Header: Shows signal strength, "Globant" network, time "19:30", and battery level "50 %".
- Chat Title: "Disaster Bot" with status "En línea" (Online).
- Message from User: "Estoy en la Región del Maule, ¿Cuál es la situación respecto a incendios forestales?" (I am in the Maule Region, what is the situation regarding forest fires?).
- Message from Disaster Bot:
  - "Es una zona de gran cantidad de incendios forestales, y con solo 5 centro de salud de urgencia."
  - "Parques como el Radal 7 Tazas se encuentra en posible riesgo, con bomberos a 70 km. y asistencia de salud de urgencia a 65 km."
  - "Muchas localidades están lejos de centros de salud, como Iloca, a más de 70 km. de la ciudad más cercana."
- Input field at the bottom.

**Screenshot 2 (Right):**

- Header: Shows signal strength, "Globant" network, time "19:30", and battery level "50 %".
- Chat Title: "Disaster Bot" with status "En línea" (Online).
- Message from User: "¿Donde sería mejor poner áreas verdes en la Región Metropolitana?" (Where would it be better to put green areas in the Metropolitan Region?).
- Message from Disaster Bot:
  - "Comuna de El Bosque, con 2 m<sup>2</sup> de áreas verdes por habitante, temperatura promedio en verano de 33.2° C. Incorporar parques podría reducir hasta en 2.8° C las temperaturas máximas."
  - "Comuna de Independencia, con 1.6 m<sup>2</sup> de áreas verdes por habitante, temperatura promedio en verano de 34.3° C. Incorporar parques podría reducir hasta en 4° C las temperaturas máximas."
- Input field at the bottom.



# Potenciales Clientes



FEMA



# ¡Muchas Gracias!

## FAQ

---



# Anexos

MA What geocoding APIs can i use to get from an address a latitud and longitude?

 There are several geocoding APIs available that can provide latitude and longitude coordinates based on an address input. Here are some popular options:

1. Google Maps Geocoding API: Google provides a comprehensive Geocoding API that allows you to convert addresses into geographic coordinates. It offers a generous free tier and supports various features, including reverse geocoding. You can find more information and access the API documentation on the Google Cloud Platform website.
2. MapQuest Geocoding API: MapQuest offers a Geocoding API that enables you to geocode addresses and retrieve latitude and longitude coordinates. It provides both forward and reverse geocoding functionality. You can refer to the MapQuest Developer Network for further details and documentation.
3. OpenStreetMap Nominatim API: Nominatim is a free and open-source geocoding service based on OpenStreetMap data. It allows you to convert addresses into coordinates and vice versa. The Nominatim API is available for public use, and you can find more information on the OpenStreetMap wiki.
4. Bing Maps API: Microsoft's Bing Maps API provides geocoding services to convert addresses into latitude and longitude coordinates. It offers a variety of geocoding options and supports multiple languages. You can explore the Bing Maps API documentation for more details.

 Regenerate response

# Anexos

NR

Ok, lets not calculate the distance, but i want to graph the closest point to a certain adress



Sure, we can mark the closest point from a given address on the map using folium. Please note that we need to convert the address to coordinates (latitude and longitude) for this purpose. Here is an example using OpenCage Geocoding API:

python

Copy code

```
from geopy.geocoders import OpenCage
from geopy.distance import great_circle
from shapely.geometry import Point

# replace 'Your OpenCage API key here' with your actual API key
geolocator = OpenCage(api_key='Your OpenCage API key here')
location = geolocator.geocode("Via Blanca 7615, Vitacura, Santiago de Chile")

# Create a Point object from the location
location_point = Point(location.longitude, location.latitude)

# Find the closest point in gdf2
gdf2['dist'] = gdf2.apply(lambda row: great_circle(location_point.coords[0],
closest_point = gdf2.loc[gdf2['dist'].idxmin()])
```

# Anexos

NR How can i extract the ino from the points

If you want to extract the information associated with each point in the GeoDataFrame, you can directly access the attributes of the DataFrame.

For example, if you want to get the disaster type, latitude, and longitude for each point, you could use the following code:

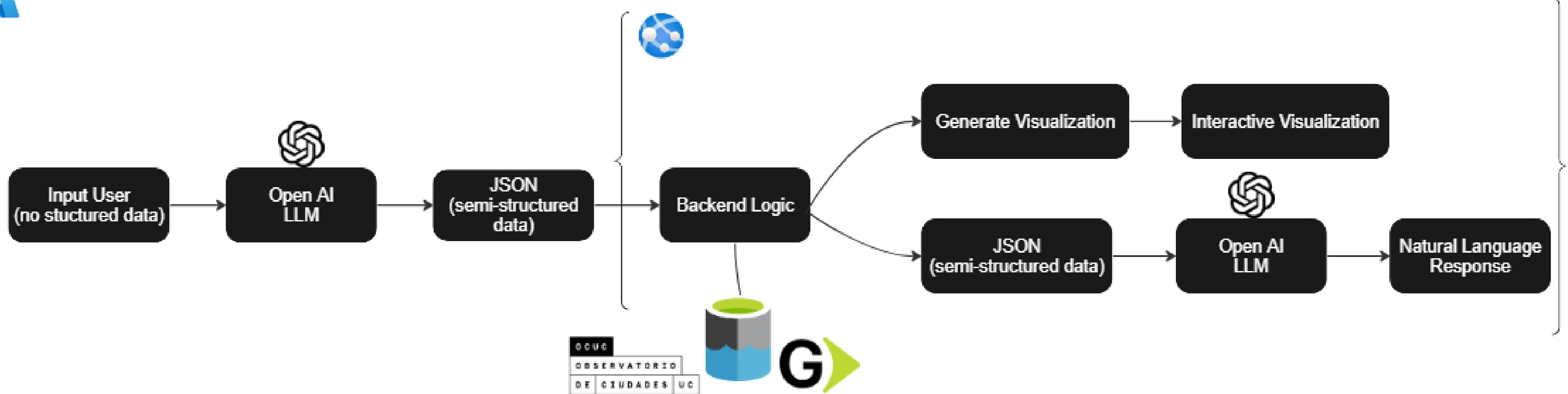
```
python
for i in range(0, len(gdf)):
    lat = gdf.iloc[i]['Latitude']
    lon = gdf.iloc[i]['Longitude']
    disaster_type = gdf.iloc[i]['Disaster Type']
    print(f"Latitude: {lat}, Longitude: {lon}, Disaster Type: {disaster_type}")
```

This will print the latitude, longitude, and disaster type for each point in your GeoDataFrame.

If you want to extract the information from the GeoJSON data, it will depend on the structure of your GeoJSON data. Typically, GeoJSON data is structured as a nested dictionary, and you can access the properties of each feature (e.g., each point or polygon) using the 'properties' key.

# Anexos

A



# Anexos