

## Microproyecto Técnico:

 Tiempo estimado: 3 horas

---

### Objetivo General

Diseñar una aplicación que:

1. Consuma datos de una API pública.
  2. Guarde los datos en una base de datos.
  3. Permite hacerle preguntas a un modelo de IA sobre esos datos extraídos de la API que se guardaron en la BD.
  4. Documentar adecuadamente.
- 

### Entregables

- Envío a correo [tpineda@oceanicsa.com](mailto:tpineda@oceanicsa.com) del Github Público del Proyecto
- Anexo de un video de **no más de 5 minutos** mostrando la funcionalidad del proyecto y explicando los puntos clave del código, de ser necesario se pedirá un espacio 1:1 para validar el proyecto realizado y el dominio sobre el mismo.

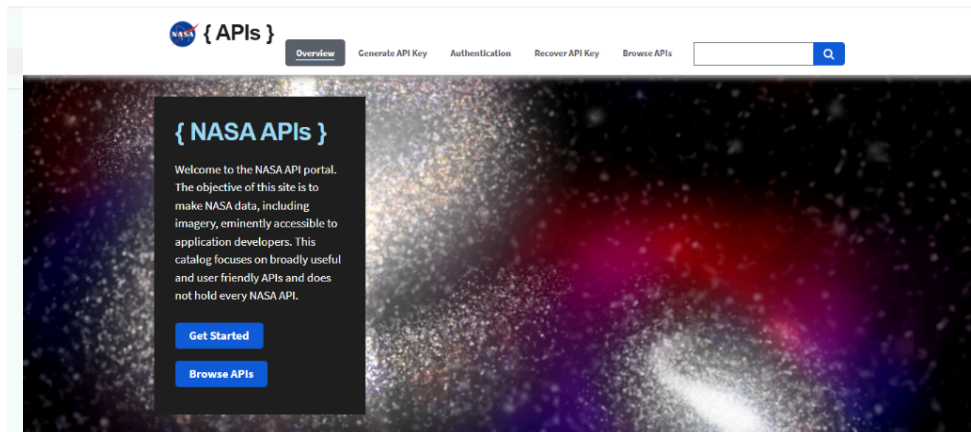
### Instrucciones

#### Paso 1. Crear Repositorio GitHub

- a. Crear Nuevo Repositorio
  - b. Nombre del repositorio: **TuNombreYApellidoOceanic2025**
  - c. Darle visibilidad pública
  - d. Clonar en editor : git clone “...”
- Hacer **commits frecuentes y claros** para evidenciar el proceso de desarrollo. Pueden ser commits por paso realizado
-

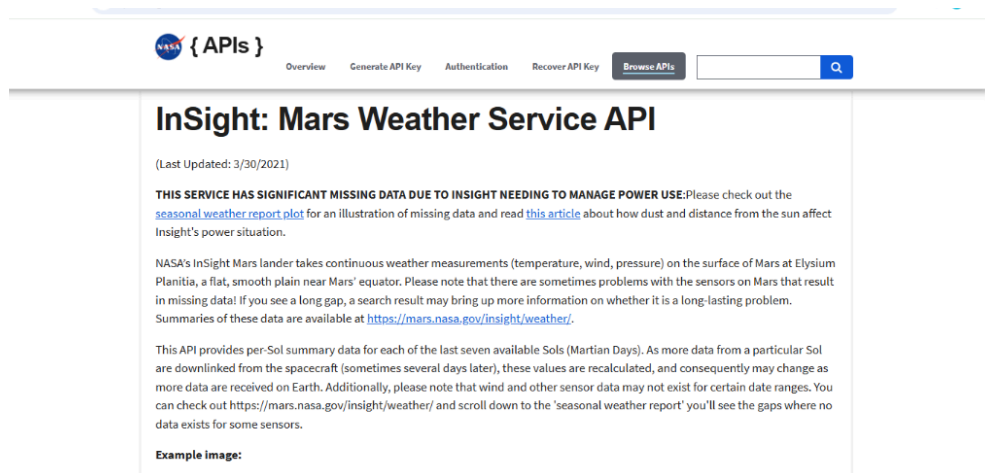
## Paso 2. Consumir una API pública

- Extraemos los datos de la NASA: <https://api.nasa.gov/>




- Debemos registrarnos en “Get Started” para recibir la API KEY a nuestro correo
- Buscaremos la API sobre datos de Marte: **InSight: Mars Weather Service API**

Este nos proveerá información de temperatura, vientos, presión por día de Marte.



Puede acceder a la documentación oficial de la API



APIs

OverviewGenerate API KeyAuthenticationRecover API KeyBrowse APIs

Q

Query Parameters

Parameter	Type	Default	Description
version	float	1.0	The version of this API
feedtype	string	json	The format of what is returned. Currently the default is JSON and only JSON works.
api_key	string	DEMO_KEY	api.data.gov key for expanded usage

Example query

[https://api.nasa.gov/insight\\_weather/?api\\_key=DEMO\\_KEY&feedtype=json&ver=1.0](https://api.nasa.gov/insight_weather/?api_key=DEMO_KEY&feedtype=json&ver=1.0)

[This document describes the API in more detail.](#)

[This self-contained HTML document](#) is an example that draws a wind rose using InSight Wind Direction frequency distribution data, which data are part of the JSON object returned by this API.

Here is something like what that wind rose looks like:

Ayuda:

API URL

[https://api.nasa.gov/insight\\_weather/?api\\_key=DEMO\\_KEY&feedtype=json&ver=1.0](https://api.nasa.gov/insight_weather/?api_key=DEMO_KEY&feedtype=json&ver=1.0)

FORMATO

```
{
  "sol_keys": [ "259", "260", "261", "262", "263", "264", "265" ],
  "259": {
    "AT": { "av": -71.233, "ct": 326642, "mn": -101.024, "mx": -27.149 },
    "HWS": { "av": 4.35, "ct": 154146, "mn": 0.150, "mx": 17.617 },
    "PRE": { "av": 761.006, "ct": 163012, "mn": 742.1498, "mx": 789.3891 },
    "WD": {
      "most_common": { "compass_degrees": 282.5, "compass_point": "SSW", "compass_right": -0.382683432365,
        "compass_up": -0.923879532511, "ct": 28551 },
      "8": { "compass_degrees": 180.0, "compass_point": "S", "compass_right": 0.0,
        "compass_up": -1.0, "ct": 17699 },
      "9": { "compass_degrees": 282.5, "compass_point": "SSW", "compass_right": -0.382683432365,
        "compass_up": -0.923879532511, "ct": 28551 },
      "18": { "compass_degrees": 225.0, "compass_point": "SW", "compass_right": -0.767106781187,
        "compass_up": -0.707106781187, "ct": 27124 }
    },
    "First_UTC": "2019-08-19T08:03:59Z", "Last_UTC": "2019-08-20T08:43:34Z", "Season": "winter"
  },
  "260": {
```

```
### Start of JSON stream
### Array of Sols with data
### Start of summary data for Sol 259
### Atmospheric temperature data for Sol 259
### Horizontal wind speed data for Sol 259
### Pressure data for Sol 259
### Wind direction summary for Sol 259
### N.B. only a subset of the WD keys are shown in this sample
### Wind direction data for most common compass point
### N.B. count (key "ct") is 28551, which is the number of
###   WD readings in this compass point
### Wind direction data for compass point 8=South; count is less
### than that for most common point; points 1-7 and 11-16 are
### excluded in this example to save space, but the counts could
### be used to display a wind rose histogram c.f. this website.
### Wind direction data for compass point 9=SSW
### N.B. count (key "ct") is 28551, which matches that of the
###   most common key above i.e. this is the same point
### Miscellaneous provance: UTC range; season.
### Start of summary data for Sol 260
### N.B. Sol 260 has no valid data for HWS, and WD has null
```

Documentación API

<https://api.nasa.gov/assets/insight/InSight%20Weather%20API%20Documentation.pdf>

Nota: El dato más importante son los días y la temperatura.

---

### Paso 3. Guardar los datos en una Base de Datos

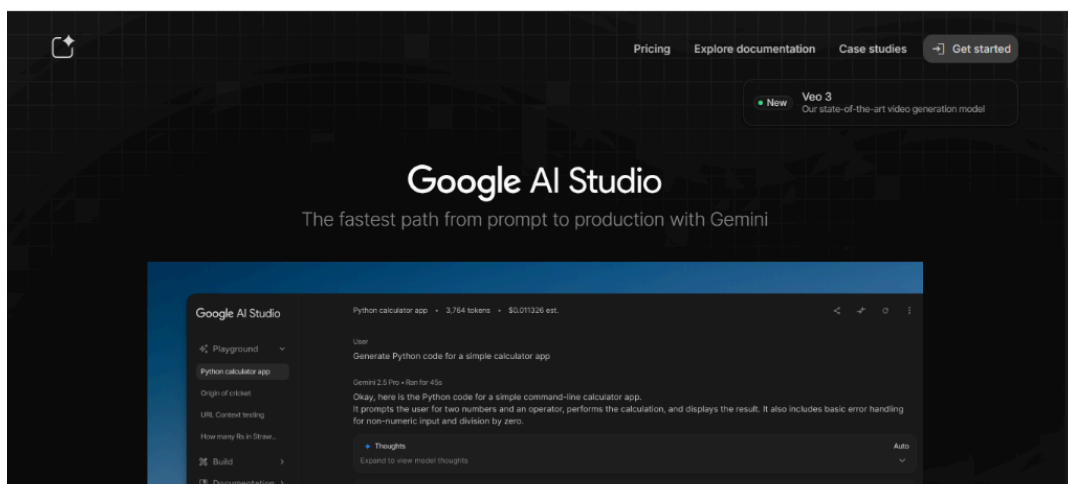
- Crear una pequeña estructura según los datos recibidos para almacenar los datos localmente.
  - Puede ser una BD temporal de poca complejidad. (ej., SQLite, una base de datos en memoria, o un archivo JSON/CSV )
  - Estos datos serán guardados y posteriormente mandados a la IA como fuente de información.
- 

### Paso 4. Hacerle preguntas a la IA

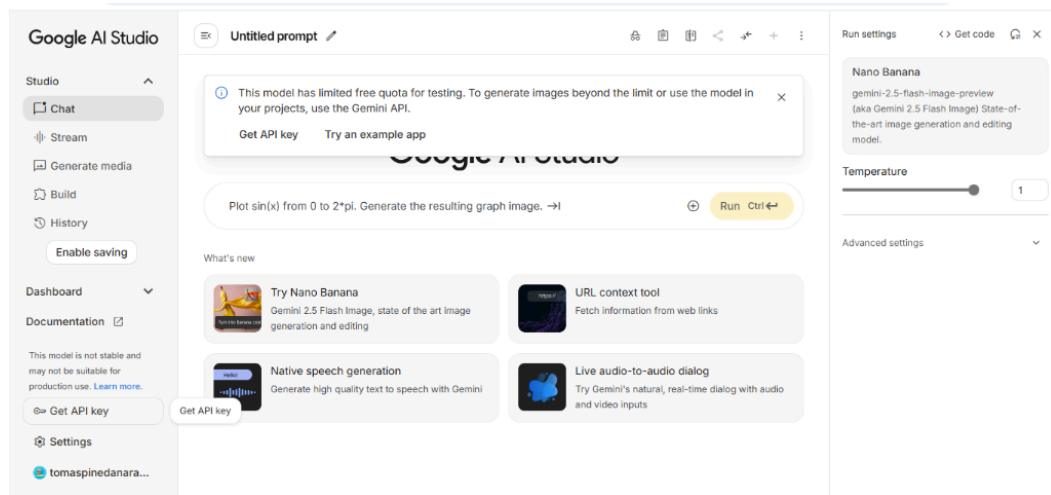
- Incluir una funcionalidad que permite hacer preguntas sobre los datos guardados, como puede ser temperatura de un día específico de Marte, temperatura máxima y mínima todo haciendo preguntas en lenguaje natural.
- Opciones posibles:
  - Usar un modelo de lenguaje pequeño (ej: GPT-2 local con Hugging Face)
  - O bien usar [langchain](#) o [llamaindex](#) con embedding básico
  - Alternativamente, usar OpenAI API, Gemini API si el candidato lo prefiera (puedes darle libertad aquí)

*Ayuda para conectarse a Gemini API:*

1. Ve a **Google AI Studio**.
2. Inicia sesión con tu cuenta de Google. **“GET STARTED”**



3. En el menú de la izquierda, haz clic en **"Get API key"** (Obtener clave de API).



4. Luego, haz clic en **"Crear clave de API"** (Crear clave de API en un nuevo proyecto).



5. Se generará tu clave. **Cópiala inmediatamente**, ya que no podrás verla de nuevo una vez que cierres la ventana. Es fundamental que la guardes en un lugar seguro.

### ***Ayuda para peticiones:***

```
curl
"https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent" \

-H 'Content-Type: application/json' \

-H 'X-goog-api-key: GEMINI_API_KEY' \

-X POST \

-d '{
  "contents": [
    {
      "parts": [
        {
          "text": "Explain how AI works in a few words"
        }
      ]
    }
  ]
}'
```

## Paso 5. Documentación

- README.md debe incluir:
    - Descripción general del proyecto
    - Tecnologías usadas
    - Funcionalidades incluidas
- 

## Paso 6. Extra

- No es obligatorio generar una interfaz para interactuar si lo considera necesario tiene la libertad de hacerlo.

## Criterios de Evaluación

Criterio		Peso
Funcionamiento general	25%	
Consumo correcto de API	15%	
Persistencia en base de datos	15%	
Capacidad de IA para responder	20%	
Buenas prácticas	15%	
Claridad del README	10%	