

POKEAPI

DOCUMENTACION

Información importante

Información

Esta es una API solo de consumo : solo el método HTTP GET está disponible en los recursos.

No se requiere autenticación para acceder a esta API, y todos los recursos son completamente abiertos y están disponibles. Desde la migración al alojamiento estático en noviembre de 2018, la limitación de velocidad se ha eliminado por completo, pero aún le recomendamos limitar la frecuencia de las solicitudes para minimizar nuestros costos de alojamiento.

Recursos de Pokémon en PokéAPI v2

1. Bayas:

- **Información general de la baya:** Nombre, ID, tamaño, suavidad, dureza, potencia, tiempo de maduración.
- **Efectos:** Lista de efectos que la baya puede tener en un Pokémon (restauración de PS, curación de estados, aumento de estadísticas en combate, etc.).
- **Firmeza:** Categoría de la firmeza de la baya (muy suave, suave, normal, dura, muy dura).
- **Sabores:** Intensidad de diferentes sabores (seco, picante, dulce, amargo, ácido).
- **Nutrientes del suelo:** Impacto de la baya en los nutrientes del suelo al ser plantada.

2. Concursos:

- **Tipos de concurso:** Nombre del tipo de concurso (belleza, dureza, ingenio, dulzura, genialidad).
- **Efectos de concurso:** Descripción de los efectos que un movimiento puede tener en un concurso.
- **Súper efectos de concurso:** Descripción de los efectos especiales de los movimientos en los súper concursos.
- **Rangos de concurso:** Diferentes rangos de los concursos (normal, súper, hiper, máximo) y sus características.
- **Tipos de movimiento por concurso:** Asociación de los tipos de movimiento con los tipos de concurso.

3. Encuentros:

- **Métodos de encuentro:** Formas en las que un Pokémon puede ser encontrado (caminando en la hierba, pescando, surf, regalos, etc.).
- **Condiciones de encuentro:** Requisitos específicos para que ocurra un encuentro (hora del día, clima, objeto equipado, etc.).
- **Valores de encuentro:** Detalles sobre la probabilidad de encuentro de diferentes Pokémon en ubicaciones específicas y bajo ciertas condiciones.

4. Evolución:

- **Cadenas de evolución:** Secuencia de evoluciones de un Pokémon, mostrando el Pokémon base y sus siguientes formas.
- **Desencadenantes de evolución:** Condiciones que deben cumplirse para que un Pokémon evolucione (subir de nivel, usar una piedra evolutiva específica, intercambiar, alcanzar cierto nivel de amistad, etc.).

5. Juegos:

- **Generaciones:** Agrupación de los juegos por generación (Kanto, Johto, Hoenn, etc.).
- **Versiones:** Nombres de las versiones de los juegos dentro de cada generación (Rojo, Azul, Amarillo, Oro, Plata, Cristal, etc.).
- **Grupos de versiones:** Agrupación de versiones que comparten características similares (por ejemplo, los juegos de la primera generación).
- **Regiones:** Las regiones geográficas donde se desarrollan los juegos (Kanto, Johto, Hoenn, Sinnoh, Unova, Kalos, Alola, Galar, Paldea, Kitakami, Blueberry Academy).
- **Índices de la Pokédex:** La Pokédex específica de cada juego y los Pokémon incluidos.

6. Elementos (Objetos):

- **Información general del objeto:** Nombre, ID, categoría, costo.
- **Efectos del objeto:** Descripción de lo que hace el objeto (curar, aumentar estadísticas, causar estados alterados, usarse para evolucionar, etc.).
- **Categorías de objetos:** Agrupación de objetos por su función (poké balls, medicinas, piedras evolutivas, objetos de combate, etc.).

7. Ubicaciones:

- **Regiones:** Las grandes áreas geográficas que contienen múltiples ubicaciones.
- **Ubicaciones:** Lugares específicos dentro de las regiones (ciudades, rutas, cuevas, edificios, etc.).

- **Áreas de encuentro:** Subdivisiones dentro de las ubicaciones donde pueden ocurrir encuentros de Pokémon.
- **Grupos de áreas de encuentro:** Agrupaciones de áreas de encuentro con características similares.

8. Máquinas:

- **Información de las Máquinas Técnicas (MT) y Máquinas Ocultas (MO):** Nombre, ID, ítem asociado (si lo hay), movimiento que enseña.
- **Grupos de MT:** Agrupación de las MT por número o por set.

9. Movimientos:

- **Información general del movimiento:** Nombre, ID, tipo, poder, precisión, puntos de poder (PP), clase (físico, especial, estado), prioridad.
- **Efecto del movimiento:** Descripción detallada de lo que hace el movimiento en combate.
- **Objetivo del movimiento:** A quién afecta el movimiento (un oponente, todos los oponentes, el usuario, etc.).
- **Grupos de movimientos:** Agrupaciones de movimientos con características similares (por ejemplo, movimientos de curación).
- **Daño por tipo:** Cómo interactúa el tipo del movimiento con los tipos del oponente.
- **Niveles de aprendizaje:** En qué niveles aprenden los Pokémon ciertos movimientos.

10. Pokémon:

- **Información general del Pokémon:** Nombre, ID, altura, peso, experiencia base, orden.
- **Sprites:** URLs de las imágenes del Pokémon en diferentes estilos (frontal, trasero, shiny, femenino, etc.).
- **Tipos:** Los tipos elementales del Pokémon (fuego, agua, planta, etc.).
- **Habilidades:** Las habilidades que el Pokémon puede tener.
- **Estadísticas base:** Los valores base de las estadísticas del Pokémon (PS, Ataque, Defensa, Ataque Especial, Defensa Especial, Velocidad).
- **Movimientos aprendidos:** Lista de movimientos que el Pokémon puede aprender y cómo los aprende (por nivel, por MT, por huevo, etc.).
- **Formas:** Diferentes formas que el Pokémon puede adoptar.
- **Especie:** Información sobre la especie a la que pertenece el Pokémon (descripciones en diferentes idiomas, grupo huevo, tasa de captura, etc.).

NOSOTROS NOS CENTRAREMOS EN PUNTO 10 EL ENDPOINT POKEMON

Datos Principales:

- **Nombre:** El nombre del Pokémon.

nombre = data['name']

- **ID:** El número de Pokédex del Pokémon.

id_pokemon = data['id']

- **Altura:** La altura del Pokémon en decímetros.

altura = data['height']

- **Peso:** El peso del Pokémon en hectogramos.

peso = data['weight']

- **Experiencia Base:** La cantidad de experiencia base que otorga este Pokémon al ser derrotado.

experiencia_base = data['base_experience']

- **Orden:** El orden para Sorting. Defecto 1 - primera forma.

orden = data['order']

- **Es Default:** Indica si esta forma es la forma por defecto para este Pokémon.

es_default = data['is_default']

Sprites (Imágenes):

- **Sprite Frontal (por defecto):** URL de la imagen frontal estándar.

sprite_frontal = data['sprites']['front_default']

- **Sprite Frontal Femenino (si existe):** URL de la imagen frontal femenina.

sprite_frontal_femenino = data['sprites']['front_female'] # Puede ser None

- **Sprite Frontal Brillante:** URL de la imagen frontal brillante.

```
sprite_fronal_brillante = data['sprites']['front_shiny']
```

- **Sprite Frontal Brillante Femenino (si existe):** URL de la imagen frontal brillante femenina.

```
sprite_fronal_brillante_femenino = data['sprites']['front_shiny_female'] # Puede ser None
```

- **Sprites de "Other" (Artwork oficial, Dream World, etc.):** Un diccionario con otras representaciones del sprite. Por ejemplo, el artwork oficial:

```
sprite_artwork_oficial = data['sprites']['other']['official-artwork']['front_default']
```

Tipos:

- **Lista de Tipos:** Una lista de diccionarios, donde cada diccionario contiene información sobre el slot del tipo y el objeto del tipo en sí (con nombre y URL). Para obtener los nombres de los tipos:

```
tipos = [tipo['type']['name'] for tipo in data['types']]
```

Habilidades:

- **Lista de Habilidades:** Una lista de diccionarios, donde cada diccionario contiene información sobre la habilidad (nombre y URL), si es la habilidad principal y su slot. Para obtener los nombres de las habilidades:

```
habilidades = [habilidad['ability']['name'] for habilidad in data['abilities']]
```

Estadísticas Base:

- **Lista de Estadísticas:** Una lista de diccionarios, donde cada diccionario contiene información sobre una estadística (nombre, URL y el valor base base_stat). Para obtener un diccionario de estadísticas base:

```
estadisticas = {stat['stat']['name']: stat['base_stat'] for stat in data['stats']}
```

Movimientos:

- **Lista de Movimientos:** Una lista de diccionarios, donde cada diccionario contiene información sobre el movimiento (nombre y URL) y las versiones en las que el Pokémon aprende ese movimiento. Para obtener una lista de los nombres de los movimientos:

```
movimientos = [movimiento['move']['name'] for movimiento in data['moves']]
```

Formas:

- **Lista de Formas:** Una lista de diccionarios con el nombre y la URL de otras formas que este Pokémon puede tener.

```
formas = [{'name': forma['name'], 'url': forma['url']} for forma in data['forms']]
```

Especies:

- **Información de la Especie:** Un diccionario con el nombre y la URL del recurso de la especie de este Pokémon.

```
especie_url = data['species']['url']
```

```
especie_nombre = data['species']['name']
```

Juegos en los que aparece:

- **Lista de Juegos:** Una lista de diccionarios que indican en qué juegos aparece este Pokémon y los encuentros específicos.

```
apariciones_juegos = data['game_indices']
```

Lista de EDNPOINT

☐ **Bayas:** /berry

☐ **Concursos:**

- Tipos de concurso: /contest-type
- Efectos de concurso: /contest-effect
- Súper efectos de concurso: /super-contest-effect

☐ **Encuentros:**

- Métodos de encuentro: /encounter-method
- Condiciones de encuentro: /encounter-condition
- Valores de condición de encuentro: /encounter-condition-value

☐ **Evolución:**

- Cadenas de evolución: /evolution-chain
- Desencadenantes de evolución: /evolution-trigger

☐ **Juegos:**

- Generaciones: /generation
- Versiones: /version
- Grupos de versiones: /version-group
- Regiones: /region
- Índices de la Pokédex: /pokedex

☐ **Elementos (Objetos):**

- Objetos: /item
- Categorías de objetos: /item-category
- Fling efectos: /item-fling-effect
- Atributos de objeto: /item-attribute

☐ **Ubicaciones:**

- Regiones: /region (se repite, ya que las regiones también agrupan ubicaciones)
- Ubicaciones: /location
- Áreas de ubicación: /location-area
- Grupos de áreas de ubicación: /location-area-encounter

☐ **Máquinas:** /machine

☐ **Movimientos:**

- Movimientos: /move
- Aprender métodos de movimiento: /move-learn-method
- Objetivos de movimiento: /move-target
- Dañar clases de movimiento: /move-damage-class
- Efectos de movimiento: /move-effect
- Cambios de valor de movimiento: /move-battle-style

□ **Pokémon:**

- Pokémon: /pokemon
- Especies de Pokémon: /pokemon-species
- Formas de Pokémon: /pokemon-form
- Hábitats de Pokémon: /pokemon-habitat
- Colores de Pokémon: /pokemon-color
- Figuras de Pokémon: /pokemon-shape
- Grupos de huevos de Pokémon: /pokemon-egg-group

Estructura general de una URL:

`https://pokeapi.co/api/v2/{endpoint}/{parametro}`

endpoint: el tipo de dato que estás consultando (como pokemon, ability, type, etc.)

parametro: el nombre o número del recurso que quieres (como "pikachu" o 25)

Endpoint	Descripción
/ability/	Devuelve información sobre habilidades pasivas que puede tener un Pokémon.
/characteristic/	Muestra qué característica define la estadística más alta de un Pokémon.
/egg-group/	Agrupar a los Pokémon que pueden reproducirse entre sí.
/gender/	Indica el género de los Pokémon y cómo se distribuyen entre especies.
/growth-rate/	Describe cómo gana experiencia un Pokémon y qué tan rápido sube de nivel.
/nature/	Cambia la forma en que se incrementan o disminuyen las estadísticas de un Pokémon.
/pokeathlon-stat/	Estadísticas usadas en los minijuegos de Pokéathlon (Gen IV).
/pokemon/	Información principal de un Pokémon: tipos, habilidades, estadísticas, sprites, etc.
/pokemon-color/	Categoriza a los Pokémon por color predominante. Usado en la Pokédex.
/pokemon-form/	Describe diferentes formas visuales de un Pokémon (ej: megaevoluciones, formas regionales).
/pokemon-habitat/	Define el hábitat natural donde normalmente viven los Pokémon.
/pokemon-location-area/	Regiones del mapa donde se puede encontrar a un Pokémon en específico.
/pokemon-shape/	Define la silueta básica del Pokémon (utilizado en la Pokédex).
/pokemon-species/	Información biológica y evolutiva de la especie Pokémon (nombre, evolución, Pokédex).
/stat/	Detalles de las estadísticas base (HP, Ataque, Velocidad, etc.).
/type/	Información sobre los tipos (fuego, agua, tierra, etc.) y sus relaciones de daño.

Ejemplos reales de URLs:

Tipo de dato	Ejemplo de URL	Qué devuelve
Pokémon	https://pokeapi.co/api/v2/pokemon/pikachu	Info completa de Pikachu
Tipo	https://pokeapi.co/api/v2/type/fire	Info de todos los Pokémon tipo fuego
Habilidad	https://pokeapi.co/api/v2/ability/overgrow	Info de la habilidad Overgrow
Especie	https://pokeapi.co/api/v2/pokemon-species/1	Info de la especie de Bulbasaur
Color	https://pokeapi.co/api/v2/pokemon-color/red	Pokémon que son rojos
Grupo huevo	https://pokeapi.co/api/v2/egg-group/monster	Pokémon que pueden cruzarse en ese grupo

Ejemplo base para hacer una petición

```
import requests

def obtener_datos(endpoint, parametro=""):
    url =
f"https://pokeapi.co/api/v2/{endpoint}/{parametro}"
    respuesta = requests.get(url)

    if respuesta.status_code == 200:
        return respuesta.json()
    else:
        print("Error al hacer la petición")
        return None
```

La relación entre la API y el problema planteado

El problema planteado consistía en desarrollar una **Pokédex interactiva** utilizando programación básica en Python, con el objetivo de consultar y mostrar información relevante de los Pokémon de forma amigable y visual. Para resolver este reto, necesitábamos una fuente confiable, completa y actualizada de datos sobre Pokémon que permitiera obtener información como: nombre, ID, tipos, altura, peso, habilidades, imágenes, formas, entre otros.

En este contexto, **la PokéAPI** (<https://pokeapi.co>) resultó ser la herramienta perfecta, ya que es una API pública gratuita específicamente diseñada para brindar acceso a toda la base de datos del universo Pokémon. Esta API cuenta con endpoints bien estructurados que nos permitieron:

- Obtener datos de **cualquier Pokémon individual** por nombre o número (usando el endpoint `/pokemon/{nombre_o_id}`).
- Consultar **todos los Pokémon de un tipo específico** mediante el endpoint `/type/{tipo}`.
- Acceder a **recursos multimedia** como imágenes oficiales en alta resolución (sprites y arte oficial).
- Trabajar con datos evolutivos, tipos, habilidades, y estadísticas.

La PokéAPI fue fundamental para que nuestro programa pudiera **consultar datos en tiempo real** y construir una experiencia dinámica y realista. En lugar de programar manualmente información de cientos de Pokémon, pudimos automatizar su recuperación desde la web con unas pocas líneas de código.

Además, su estructura en formato **JSON** facilitó el procesamiento de los datos dentro de Python, permitiéndonos organizar la información en tarjetas, realizar búsquedas dinámicas, aplicar filtros por tipo y construir una navegación fluida tipo “Pokédex” como la que se ve en los videojuegos originales.

En resumen, **la PokéAPI resolvió directamente nuestro problema de acceso a datos confiables y completos sobre Pokémon**, y se integró perfectamente con Python y las librerías usadas (como requests, Pillow y tkinter). Gracias a ella, pudimos convertir una idea académica en una aplicación funcional e interactiva, fortaleciendo nuestras habilidades tanto técnicas como creativas en programación.