

# Prueba

---

Para desarrollar esta prueba nos conectaremos a la [API de la NASA](#) para descargar fotos de los Rovers y filtrar resultados acorde a lo pedido.

## ¿Qué tenemos que saber para enfrentarnos a esta prueba?

- Consumo de una API REST con autenticación sencilla
- Crear métodos que reciban parámetros
- Iterar resultados dentro de un diccionario y una lista
- Filtrar resultados dentro de un diccionario y una lista
- Guardar los resultados dentro de un archivo

## Requerimientos

---

1. Una función `request` que debe:

- Recibir como parámetro la URL y el `api_key`
- Concatenar la API key en la URL
- Retornar el diccionario con los resultados

**Tip:** Se puede seguir el paso a paso explicado en la lectura en el capítulo API con autenticación e ingresar al [Portal de la Nasa](#) para autenticarse.

2. Una función llamada `build_web_page` que debe:

- Recibir como parámetro el diccionario de respuesta con todos los datos de las fotos.
- Construir una página web que muestre las fotos.

## La estructura del documento generado debe seguir el siguiente formato:

---

```
...
<html>
<head>
</head>
<body>
<ul>
  <li><img src='../398380645PRCLF0030000CCAM04010L1.PNG'></li>
  <li><img src='../398381687EDR_F0030000CCAM05010M_.JPG'></li>
</ul>
</body>
</html>
...
```

- **Tip:** Al revisar la documentación, se encontrarán con un parámetro que representa la cantidad de fotos. No coloquen un valor muy alto, o pueden generar una página web que tarde mucho en cargar.
- **Pregunta bonus:** Crear la función `photos_count` que reciba el diccionario de respuesta, y devuelva un nuevo diccionario con el nombre de la cámara y la cantidad de fotos.