

## Desafío - Arreglos, hashes y APIs

En este desafío validaremos nuestros conocimientos de Arreglos, Hashes y consumo de APIs con Ruby..

Lee todo el documento antes de comenzar el desarrollo **grupal**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Tiempo asociado: 2 horas cronológicas.

### Descripción

Aplicando los conceptos y herramientas aprendidas hasta ahora 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 este desafío?

- Consumo de una API Rest con autenticación sencilla.
- Crear métodos que reciban parámetros.
- Iterar resultados dentro de un hash y un arreglo.
- Filtrar resultados dentro de un hash y un arreglo.
- Generar un archivo HTML con los resultados de la consulta a la API.

Link de la API a consultar

- [https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&api\\_key=DEMO\\_KEY](https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&api_key=DEMO_KEY)



**Nota:**

Esta API requiere autenticación simple para conectarse y de ese modo obtener un token de acceso. En el link entregado ya viene con ese `api_key` generada solo para visualizar la información que contiene. **Deberás generar tu propio** `api_key` siguiendo el proceso indicado en su [documentación](#).

## Pasos para acceder a Mars Rover Photos

Una vez generado el api\_key deberás ingresar en el menú los siguientes apartados:

1. En el menú de navegación ingresar a **Browse APIs**.
2. Luego, buscar en el listado de información **Mars Rover Photos**.
3. Luego en **Example Queries**, seleccionar el siguiente enlace

```
https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000  
&api_key=DEMO_KEY
```

4. En el parámetro `?sol=` aparece el número `1000`, esto retorna mil fotos asociadas a la consulta, modifica el número e ingresa al menos `10` para que el retorno de información no sea tan extenso.
5. En el parámetro `$api_key=`, deberás ingresar la key generada durante el registro solicitado en la documentación.

## Requerimientos

1. Crear el método request que reciba una url y retorne el hash con los resultados.  
(4 Puntos)
2. Crear un método llamado `buid_web_page` que reciba el hash de respuesta con todos los datos y construya una página web. Se evaluará la página creada y tiene que tener este formato: (5 Puntos)

```
<html>  
<head>  
</head>  
<body>  
<ul>  
  <li><img src='.../398380645PRCLF0030000CC AM04010L1.PNG'></li>  
  <li><img src='.../398381687EDR_F0030000CCAM05010M_.JPG'></li>  
</ul>  
</body>  
</html>
```

3. Crear un método `photos_count` que reciba el hash de respuesta y devuelva un nuevo hash con el nombre de la cámara y la cantidad de fotos. (1 Punto)



¡Mucho éxito!

## Consideraciones y recomendaciones

- En la guía de ejercicios del Bloque 11 tienes documentación de cómo generar un HTML a partir de datos obtenidos en una API.
- Recuerda que los hashes están compuestos por `clave: valor`.
- Una forma rápida de invertir un hash es con el método `.invert()`.
- Recuerda que puedes generar el código de Ruby para la consulta directamente desde Postman.