

## Taller 2

### API REST

#### Objetivo general

El objetivo de este taller es practicar los conceptos básicos de API REST revisados en la clase anterior. Cada ejercicio le permitirá poner en práctica los conceptos vistos en las diapositivas anteriores. Recuerde poner en práctica las buenas prácticas de programación discutidas en el curso. Comente y ponga nombres significativos a las funciones y variables para que su código sea claro.

#### Punto 0 | Preparación

Para este taller vamos a utilizar la herramienta Postman ([Download Postman | Get Started for Free](#)) esta nos permite hacer pruebas sobre un API cualquiera. La vamos a utilizar tanto en unas APIs de uso público como en la que nosotros mismos creemos. Retome las clases trabajadas en los talleres anteriores (Todos los animales).

#### Parte 1 | APIS públicas

En primer lugar, vamos a jugar un poco con tres APIs que se encuentran en internet. Para cada una vamos a hacer una petición utilizando la página web y luego otra utilizando postman luego de leer la documentación. Cree un archivo en word para guardar las capturas de pantallas que vaya tomando en esta parte del taller.

#### Punto 1 | Perros

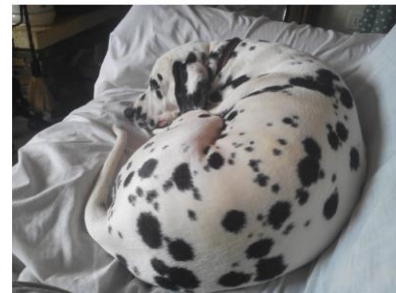
Dog API es un API que permite consultar fotos de perros. (<https://dog.ceo/dog-api/>). Al entrar a la página podrá consultar una foto aleatoria de un perro:

`https://dog.ceo/api/breeds/image/random` [Fetch!](#)

JSON

```
{
  "message": "https://images.dog.ceo/breeds/dalmatian/cooper1.jpg",
  "status": "success"
}
```

IMAGE



Guarde una captura de alguna foto que le guste. Lea la documentación disponible en la página web y por medio de postman realice una petición de **3 perros de raza Bernés de la montaña**. Postman devolverá un JSON con el resultado de la consulta, abra los links de las 3 fotos y coloque captura de pantalla de los 3 perros devueltos en el documento. **Pista:** Bernés es una subraza de la raza de Montaña.

## Punto 2 | Pokémon

PokeAPI es un API que permite consultar toneladas de información sobre diferentes Pokémon (<https://pokeapi.co/>). En la página principal realice una consulta sobre Pikachu, así como se ve en el siguiente pantallazo. A continuación, busque la parte de la respuesta en que se muestre la parte de la respuesta que muestra los **sprites**, es decir las fotos utilizadas dentro del juego por el pokémon. Esto se verá como una lista de links de imágenes.



A continuación, lea la documentación y por medio de Postman cree una consulta en la cual le preguntemos al API acerca de la **naturaleza “Calmado” (Calm)** y tome una captura de pantalla en que muestre cuál es el **sabor que odian los Pokémon de esta naturaleza**.

## Punto 3 | Rick and Morty

El API de Rick and Morty (<https://rickandmortyapi.com/>) contiene información sobre episodios, personajes, localizaciones y otra información relacionada con la famosa serie. En este caso no tenemos cómo probarlo desde la página web, así que haremos directamente la consulta en Postman.

Realice una consulta de **las localizaciones cuyo nombre incluye la palabra “Dorian”** y a continuación tome una captura de pantalla de la lista de habitantes del planeta Dorian 5.

## Parte 2 | Nuestra API

El siguiente punto es crear nuestra propia API. Utilizaremos las clases Gato, Perro, Hurón y Boa Constrictor. El API solo responderá a una consulta.

## Punto 4 | Creación de Repositorio

Cree un repositorio en Github y cargue en este su trabajo de este proyecto. Coloque como nombre TALLER2-LOGIN donde login es su login uniandes. No utilice puntos.

Remítase al taller 3 del módulo 2 o a las diapositivas, si no está seguro de cómo crear el proyecto.

### **Punto 5 | Nuestro propio API, ¿cómo hace el animal?**

Construya un API utilizando las funciones y las clases disponibles. De manera que se pueda consultar cómo hacen los 4 animales diferentes (Gato, Perro, Hurón, Boa constrictor). **Tiene libertad para escoger el nombre del endpoint.** Tome capturas de pantalla de la respuesta de su API sobre cómo hacen el Perro y el Hurón, estas consultas deben hacerse desde Postman. No olvide también crear la vista web para la consulta, en caso de que alguien visite la página web y revise estos endpoints.

### **Entrega**

Entregue el vínculo a su repositorio con el trabajo desarrollado través de Bloque Neón en el taller del módulo 3 designado como “Taller 2: API REST”.