# Tarea 2

Hola a todos! les dejo la tarea para que la revisen. Recuerden que la tarea va con algunas 3 décimas para la evaluación final.

## IMPORTANTE

La tarea deben desarrollarla en su repositorio personal, si no saben ocupar git para subirlo, me comentan y les ayudo. Me deben compartir el enlace de su repositorio con la tarea, no acepto .zip o algún otro formato ya que me toma más tiempo revisarlo ya que debo descargarlo. Solo aceptaré enlaces al repositorio.

### Ejercicios

1. Crear un objeto, agregar como sus propiedades lo siguiente:
   * Una propiedad de tipo **arreglo** llamado mascotas que incluya estos elementos: 'perros','gatos','canarios','pez','serpiente'
   * Una propiedad de tipo **function** llamada eliminarMascota, que reciba el 1 parámetro como el tipo de la mascota a eliminar
     + Esta función debe buscar según el parámetro dentro del arreglo definido anteriormente llamado mascotas y eliminar el tipo de mascota
     + Esta función debe retornar el arreglo actual

Tip / Ayuda

* + Una propiedad de tipo **function** llamada eliminarUltimaMascota, que no reciba parámetros y que al ejecutarse:
    - Esta función debe eliminar el tipo de mascota al final del arreglo definido anteriormente llamado mascotas
    - Esta función debe retornar el arreglo actual
  + Una propiedad de tipo **function** llamada agregarMascota, que reciba el 1 parámetro como el tipo de nueva mascota a agregar
    - Esta función debe agregar la nueva mascota al final del arreglo definido anteriormente llamado mascotas
    - Esta función debe retornar el arreglo actual
  + Una propiedad de tipo **function** llamada contadorMascotas, que no reciba parámetros y que al ejecutarse:
  + Esta función debe contar dentro del arreglo cuantos tipos de mascotas en su definición terminan en os, por ejemplo Perros termina en os
    - Guardar el resultado en una variable
    - La función debe retornar un string o texto, que diga En el arreglo hay 3 mascotas que terminan con 'os'

1. Dado el siguiente objeto:
2. const trabajador = {
3. nombre: 'Jhon Smith',
4. cargo: 'QA',
5. empresa: {
6. ubicación: {
7. comuna: 'Santiago',
8. puesto: 'nº 24',
9. },
10. datos: {
11. nombre: 'ACME',
12. }
13. clientes: ['Facebook', 'Google'],
14. }
15. gustos: ['comer', 'ver televisión', 'dormir']
16. hijos: null,

}

Deben generar las siguientes funciones, **deben ocupar desestructuración**, todos los que estan en parentesis deben venir desde la información del objeto:

* + Hacer una función que reciba el objeto trabajador y devuelva un mensaje: El trabajador (Jhon Smith) trabaja en (ACME) con cargo (QA) y le gusta (comer) y (2) más, (no tiene hijos)
  + Hacer una función que reciba el objeto trabajador y devuelva un mensaje: El trabajador (Jhon Smith) va a su trabajo en (Santiago), es (QA), en el puesto (nº 24), trabaja con (Facebook) y (Google)