

## Test práctico

A continuación se detallan 5 problemas. Se deben entregar mínimo 4 de los 5 problemas. Tomar en cuenta las siguientes pruebas.

### Consideraciones generales

- Crear un repositorio en Github / Gitlab o Bitbucket. Debe ser un repositorio privado.
- El código debe poder ser ejecutado con versiones vanilla del lenguaje a utilizar.
- Seguir estándares de código como si se fuera entregar a un cliente.
- Simplicidad en la lógica utilizada.

La implementación puede ser tan sencilla o complicada como desees, realiza la tarea de un modo que expongas de la mejor manera tus conocimientos.

### 1. Fetch & order

Usando <https://randomuser.me/> como fuente de data, crear una función que retorne un arreglo:

- Debe retornar 10 personas.
- Las personas deberán estar ordenadas por primer nombre.

### 2. Fetch & find

Usando <https://randomuser.me/> como fuente de data, crear función, que acepte como argumento una edad y retorne la data de la persona:

- Retornará 1 sola persona.
- La persona a retornar será mayor de la edad especificada como argumento.

### 3. Fetch & count

Usando <https://randomuser.me/> como fuente de data, crear función, que retornará un char/string:

- Deberá obtener 5 personas.
- En base a los nombres deberá calcular cual es la letra más utilizada en los nombres completos de las 5 personas.

### 4. Fastest ship

Usando <https://swapi.dev/> como fuente de data, crear función, que retornará un string:

- Aceptara un argumento tipo entero, que indicará la cantidad de pasajeros requerida.
- Debera calcular sobre todas los starships
- Retornara el nombre la nave que coincida con los siguientes parámetros:
  - Tiene la capacidad para transportar los pasajeros indicados
  - Puede viajar por al menos 1 semana.
  - Fue parte de la trilogía original (4, 5, 6)
- Si más de una nave coincide con dichos parámetros, debera retornar la mas rapida.

## 5. Planet by terrain

Usando <https://swapi.dev/> como fuente de data, crear función, que retornará un string:

- Aceptara un argumento tipo string, que indicará el tipo de terreno.
- Retornara el nombre del planeta que coincida con los siguientes parámetros:
  - Coincide el tipo de terreno especificado como parámetro.
- Si más de un planeta coincide con dichos parámetros, deberá retornar el que posea más población.