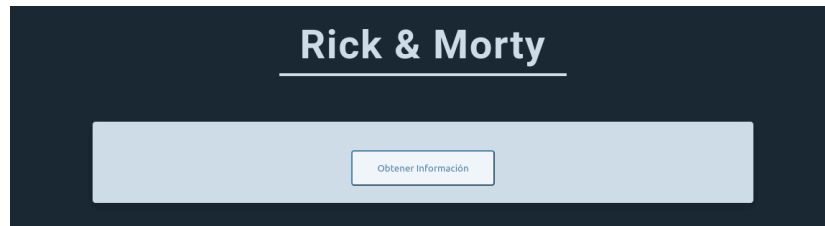


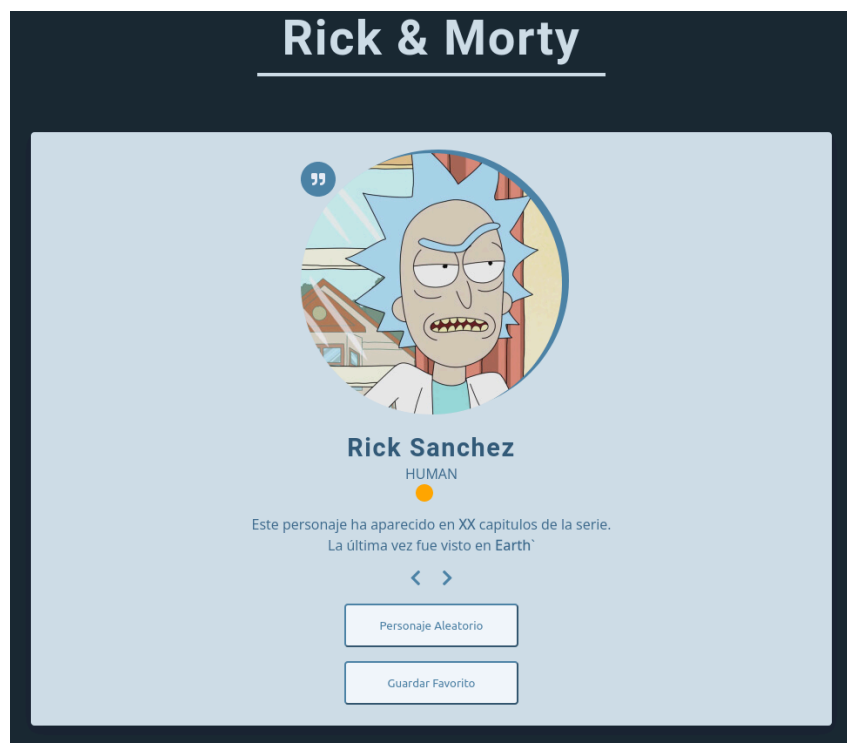
## DWEC04 - Tarea 01

### ENUNCIADO DE LA TAREA:

En la plantilla que se os facilita aparece una sección vacía (al principio) con un botón para obtener información acerca de los personajes de la serie “Rick&Morty”.



Cuando pulsemos el botón tendremos que visualizar la información de personajes de la serie, quedando algo como:



Los datos que tienen que aparecer son:

- Nombre.
- Especie.
- Estado (en formato de punto con un color, hay 3 posibilidades)
- Información redactada del personaje con información de:
  - En cuantos capítulos aparece el personaje.
  - Cual ha sido el último lugar donde se le ha visto (location)

Cuando estemos visualizando la información de los personajes, dispondremos de **4 botones** adicionales:

- “<”, para pasar al personaje anterior (si es el primero, pasará al último)
- “>”, para pasar al personaje siguiente (si es el último, pasará al primero)
- “Personaje Aleatorio” carga aleatoriamente información de otro usuario de la serie.
- “Guardar Favorito”, que significa que tendremos que crear una cookie especial en el navegador.

Esta cookie es especial ya que tendremos **que comprobar si existe o no cuando inicie la página**. En caso de existir se cargará directamente la información de ese personaje en la página (no tendremos que pulsar el botón inicial). La cookie tendrá una vigencia de **5 min**. Si tras ese periodo abrimos la página tendremos que pulsar el botón de obtener información para cargar los datos de los personajes.

### ACLARACIONES EL ENUNCIADO:

- Para obtener información de los personajes utilizaremos la API de Rick&Morty ([enlace](#)). Tenéis unos ejemplos de cómo se utilizan en la documentación.
- Hay varias (bastantes) páginas de personajes, con utilizar la primera será suficiente (20 personajes). Si alguien quiere hacerlo con todas las páginas tiene vía libre.
- Los HTML son a modo de ejemplo. No tenéis que tocar nada más en ellos, solo fijaros en la estructura de los elementos (para que el CSS lo visualice como en las fotos que adjunto).
- Podéis utilizar **XHR** o **fetch** u otra librería (jQuery, Axios, ...) para la obtención de los datos, pero **será requisito utilizar async/await** en algún lugar del código.
- El único cambio que necesitáis hacer en el HTML es añadir la referencia al script externo. Cuando entreguéis la tarea solo deberéis entregar un documento “index.html”.

### ACLARACIONES SOBRE LA ENTREGA DE LA TAREA:

- Para la entrega de esta tarea comprime todos los documentos que forman el proyecto (\*.html, \*.js, ...) en un solo fichero. El archivo se nombrará siguiendo las siguientes pautas:

**Apellido1\_Apellido2\_Nombre\_DWEC04\_Tarea01**

- Utilizar los comentarios de código para realizar breves explicaciones. En caso de necesitar/querer explicar más en detalle el ejercicio utilizar la [plantilla](#) e ir adjuntando capturas de pantalla del código y las explicaciones y aclaraciones necesarias .
- Criterios de evaluación relacionados con la tarea:

RA-3	CE	g) Se han utilizado "cookies" para almacenar información y recuperar su contenido.
RA-7	CE	a) Se han evaluado las ventajas e inconvenientes de utilizar mecanismos de comunicación asíncrona entre cliente y servidor web.
	CE	b) Se han analizado los mecanismos disponibles para el establecimiento de la comunicación asíncrona.
	CE	c) Se han utilizado los objetos relacionados.
	CE	d) Se han identificado sus propiedades y sus métodos.
	CE	e) Se ha utilizado comunicación asíncrona en la actualización dinámica del documento web.
	CE	f) Se han utilizado distintos formatos en el envío y recepción de información.
	CE	g) Se han programado aplicaciones web asíncronas de forma que funcionen en diferentes navegadores.
	CE	h) Se han clasificado y analizado librerías que faciliten la incorporación de las tecnologías de actualización dinámica a la programación de páginas web.
	CE	i) Se han creado y depurado programas que utilicen estas librerías.