



# Ejercicios React

**Prácticas con componentes, ciclo de vida del componente, eventos etc.**

1. Hello world - *Dificultad:* ●

Crear una aplicación web con un componente que muestre en pantalla el mensaje “Hello World”.



2. Hello world (parte 2)- *Dificultad:* ●

Modificar el componente anterior y enviar mediante props el valor “My friend” de manera que el mensaje quede de la siguiente forma:



3. Hello world (parte 3) - *Dificultad:* ●

Modificar el programa anterior agregando al componente un state con la clave msj: “(from changed state)”, además agregar un botón al presionar este botón (onClick), el state de la clave msj se debe mostrar por pantalla de la siguiente manera:



#### 4. Lista de tareas - *Dificultad:* ●

Crear una aplicación web que muestre una lista de tareas, al ingresar una tarea en el input y presionar enter, esta tarea se debe mostrar en una lista como la imagen a continuación:

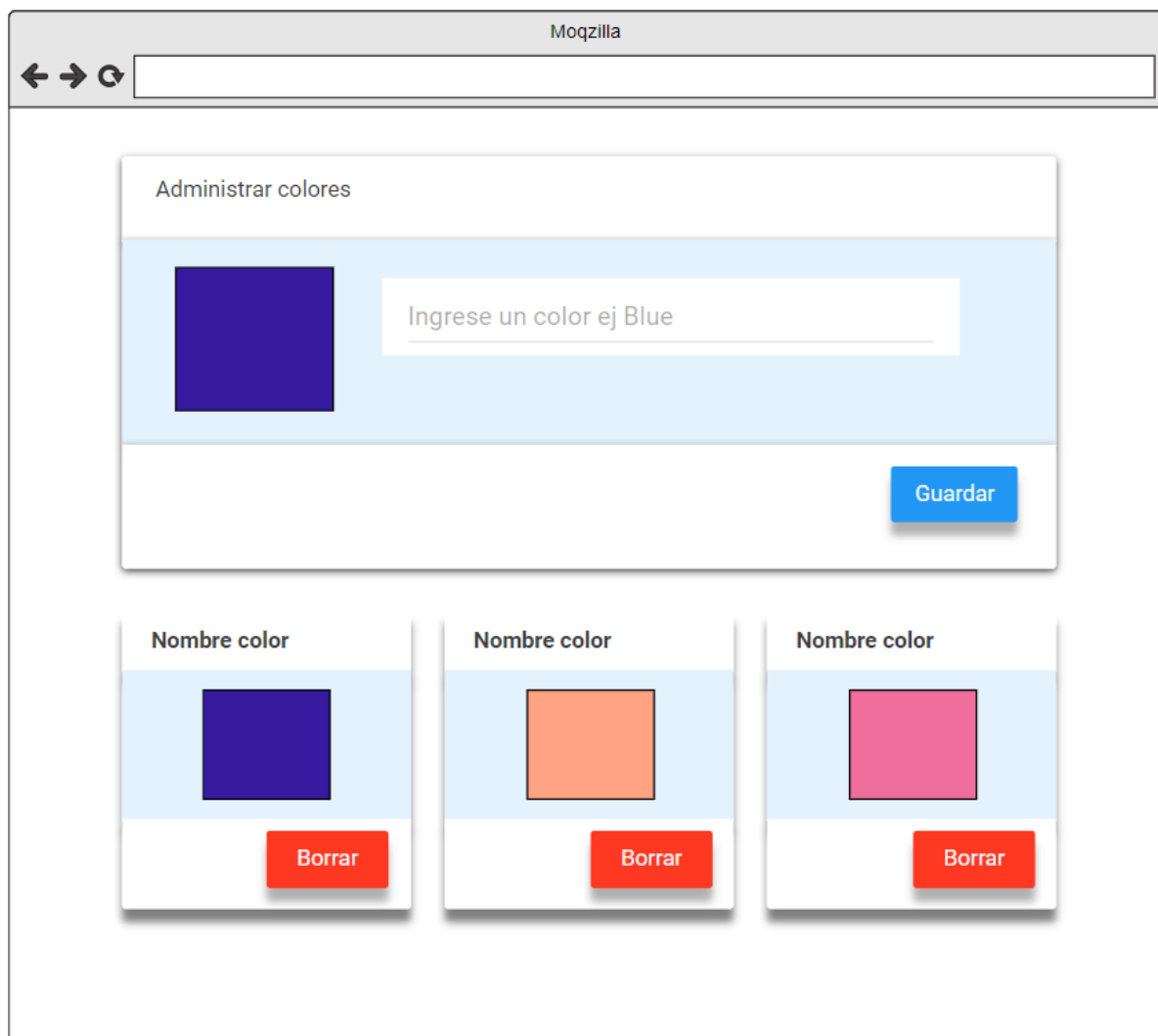


#### 5. Lista de tareas (con localStorage)- *Dificultad:* ●●

Modifica el programa anterior para que las tareas se puedan almacenar en el localStorage.

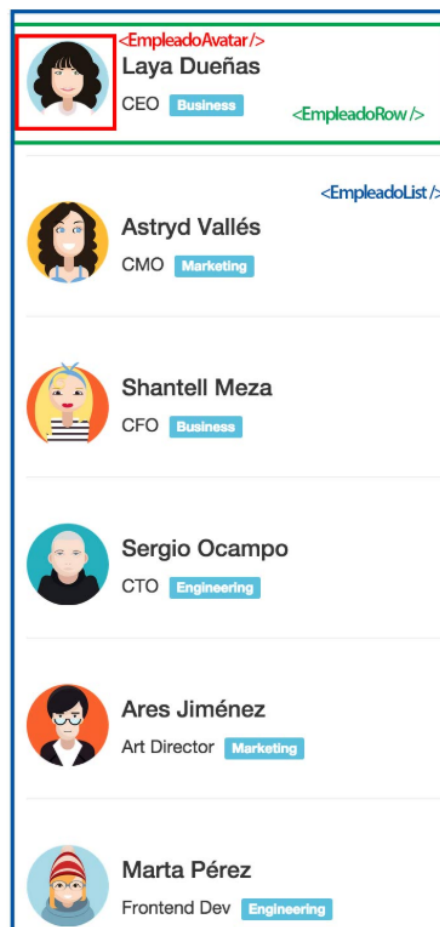
## 6. Paleta de colores- *Dificultad:* ●●

Crear un proyecto de react que muestre un formulario con un input y un button que permite guardar un color en el localStorage, similar a la siguiente imagen:



## 7. Lista de empleados - *Dificultad:* ●●

Crear una aplicación web que permite reutilizar componentes siguiendo la estructura de la siguiente imagen, pueden incorporar bootstrap u hojas de estilo propias.



Tendremos un EmpleadoAvatar (recuadro rojo), un EmpleadoRow (recuadro verde) y por último un EmpleadoList (recuadro azul).

Los datos de los empleados deberán ser mostrados mediante un arreglo que deben crear similar al siguiente:

```
let empleados = [
  { id: 1, fullName: "Laya Dueñas", title: "CEO", department: "Business",
    pic: "empleado01.png" },
  { id: 2, fullName: "Astryd Vallés", title: "CMO", department:
    "Marketing", pic: "empleado02.png" },
  { id: 3, fullName: "Shantell Meza", title: "CFO", department:
    "Business", pic: "empleado03.png" },
  { id: 4, fullName: "Sergio Ocampo", title: "CTO", department:
    "Engineering", pic: "empleado04.png" },
  { id: 5, fullName: "Ares Jiménez", title: "Art Director", department:
    "Marketing", pic: "empleado05.png" },
  { id: 6, fullName: "Marta Pérez", title: "Frontend Dev", department:
    "Engineering", pic: "empleado06.png" },
]
```

```
{ id: 7, fullName: "Ellen Balderas", title: "Digital Strategist",  
department: "Marketing", pic: "empleado07.png" },  
{ id: 8, fullName: "Cynthia Valentín", title: "Backend Dev", department:  
"Engineering", pic: "empleado08.png" },  
{ id: 9, fullName: "Bernard Jung", title: "DevOps Engineer", department:  
"Engineering", pic: "empleado09.png" },  
]
```

## 8. Formulario simple - Dificultad:

Crear una aplicación web con react que tenga los siguientes componentes:

1. Título “Formulario” con un color de background diferente al color del body.
2. Formulario, con los campos de Nombre, Apellido, DNI, email y botón que me permita enviar el formulario.

Una vez presionado el botón de enviar se debe verificar que todos los campos fueron cargados, y mostrar un alert con el mensaje “datos enviados”, si los datos están incompletos mostrar un alert con el mensaje “Completar todos los datos”.

## 9. Pacientes de veterinaria - Dificultad:

Crear una aplicación web que permite cargar un formulario para registrar una cita en una veterinaria, todos los datos solicitados en el formulario deben ser obligatorios.

La aplicación debe permitir borrar las citas y estas deben ser guardadas en el localStorage.

Mozilla

← → ↻

## Administrador pacientes de veterinaria

Llenar el formulario para crear una cita

Nombre de mascota:	<input type="text" value="nombre de mascota"/>		
Nombre de dueño:	<input type="text" value="nombre de dueño"/>		
Fecha	<input type="text" value="dd/mm/yyyy"/>	Hora:	<input type="text" value="hh:mm"/>
Sintomas:	<input type="text" value="describir sintomas"/>		

Agregar nueva cita

No hay citas

Si hay citas debo mostrarlas, con sus respectivos datos dentro de una card:

## Administra las citas aquí

Mascota: Firulais  
Dueño: Pepe

Fecha:

Hora:

Síntomas:

Borrar

Mascota: pompon  
Dueño: rita

Borrar

### 10. Alta de Películas - Dificultad: ●●●

Crear una aplicación web con react que contenga un formulario con los siguientes campos:

- nombre (en un campo de texto)
- descripción (en un textarea)
- género (en un select donde podemos seleccionar comedia, drama o infantil)

Cuando presione enviar, guardar esta película en el localStorage y mostrar en una card.

### 11. Web de noticias - Dificultad: ●●●

Crear una aplicación web con react, que consuma la API provista por <https://newsapi.org/> o <https://newsdata.io/docs>, la aplicación debe tener la siguiente estructura:





Componentes:

1. Título
2. Formulario (este contiene el select)
3. ListaNoticias
4. Noticia

Desde el select superior, debe poder elegir las diferentes categorías (provistas por la api ver en la documentación) y cargar en cards las mismas.

NOTA: la api newsapi funciona solo en forma local y no en un proyecto de producción, si quieren deployar el proyecto en netlify usar newsdata

12. Web de noticias v2 - Dificultad: ●●●

Modifica la web anterior permitiendo que el usuario seleccione el país y la categoría de las noticias que quiere ver.

13. Web del clima - Dificultad: ●●●●

Crea una aplicación web, que permita escribir una ubicación y un país, luego presione un botón consultar y muestre el clima del lugar ingresado, consultar el clima utilizando la API <https://openweathermap.org/>. Si la ubicación ingresada no

existe, mostrar un cartel de error indicando que no se encontraron datos de la ciudad ingresada.

#### 14. Blog de recetas - Dificultad:

Crea una aplicación web, que permita mostrar un blog de recetas de cocina en una página, desde otra pagina debo poder agregar, modificar o quitar las recetas y estas deben estar almacenadas en el localStorage o json-server.