

## UD 08. REACT

Desarrollo Web en entorno cliente  
CFGS DAW

## Ejercicios

Álvaro Maceda Arranz  
[alvaro.maceda@ceedcv.es](mailto:alvaro.maceda@ceedcv.es)

2022/2023

Versión:230115.1747

## Licencia




**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

## ÍNDICE DE CONTENIDO

<b>1. Ejercicio 1: Gradient.....</b>	<b>3</b>
<b>2. Ejercicio 2: Picker.....</b>	<b>4</b>
<b>3. Ejercicio 3: SuperPicker.....</b>	<b>4</b>

## UD08. REACT

### 1. EJERCICIO 1: GRADIENT

Crea un componente **React** llamado **Gradient** que admita tres propiedades:

- **hue**: Valor entre 0 y 360 que indicará el matiz (hue) en un modelo de color HSL
- **saturation**: Valor entre 0 y 100 que indicará la saturación en un modelo de color HSL
- **number**: Indicará el número de cuadrados a crear en el componente

El componente creará tantos **divs** como los indicados en **number**. Los **divs** tendrán el color **hsl(hue, saturation, lightness)** donde **hue** y **saturation** serán las pasadas como parámetro y **lightness** se incrementará desde 0 a intervalos de **100/number**.

Por ejemplo, si se muestra el componente con:

```
<Gradient hue={23} saturation={100} number={5}/>
```

Se mostraría esto en pantalla:



Los colores de los **divs** serían:

- **hsl(23, 100%, 0%)**
- **hsl(23, 100%, 20%)**
- **hsl(23, 100%, 40%)**
- **hsl(23, 100%, 60%)**
- **hsl(23, 100%, 80%)**

Puedes utilizar este CSS para los elementos HTML:

```
.squares {  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: center;  
  align-items: center;  
  height: 100%;  
}  
  
.square {  
  height: 100px;  
  flex-grow: 1;  
}
```

📖 En los ejercicios puedes crear tantos componentes React como creas necesario para realizar la tarea; normalmente necesitarás utilizar más de uno.

## 2. EJERCICIO 2: PICKER

Crea un componente **React** llamado **Picker** que tenga este aspecto:

Hue:  0

Saturation:  100

Steps:  10

Hue: 0

Saturation: 100

Steps: 10

El componente tendrá tres sliders creados con `<input type="range">`. El primero tendrá valores entre 0 y 360, el segundo entre 0 y 10 y el tercero entre 5 y 100. Al cambiar los valores de los sliders se mostrará el valor de los mismos en el recuadro inferior.


## 3. EJERCICIO 3: SUPERPICKER

Modifica el componente del ejercicio 2 para que, en vez de los valores, se muestre un degradado utilizando el componente del ejercicio 1:

Hue:  0

Saturation:  100

Steps:  10



*A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento:*

- *Álvaro Maceda Arranz*