

COLOR

UNCPBA/ FCE / 2020
INTERFACES DE USUARIO E INTERACCIÓN | TUDAI

Administrativo

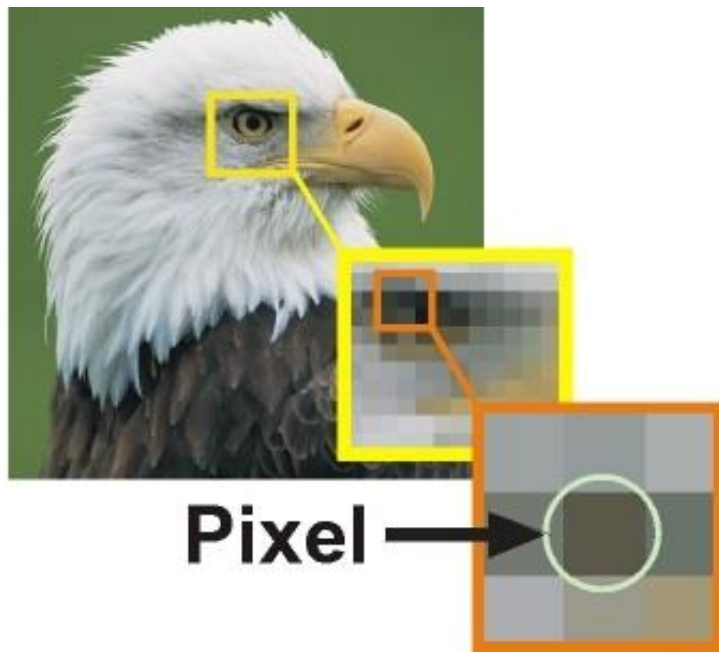
- Listado de LU-Mail completo → **éxito**. Si queda alguien: mail a mí.
- Creación del grupo con mails oficiales: DONE.
- Uso del grupo y reglas: DONE.
- Subida de clases grabadas a la web: DONE.

Administrativo [PROG2]

- La verdad q debiera ser correlativa, es un tema q ya estamos tratando para la próxima versión del plan “2.0”.
 - Revisamos los horarios intra-cátedra y con cátedra de Prog2. Conclusión: No podemos moverlo.
 - **LA BUENA:** Tienen clases grabadas, podemos ajustar algunas explicaciones para que no se la pierdan. También consulta por el grupo! 😊
 - El plan de “cambio de horario ok, pero con un costo” → experimento fallido.
NO SE PUEDE.
-
- Coordinación TUDAI, espacio de cátedra

Imágenes [Repaso]

Una imagen digital es una matriz 2D (WxH) de píxeles.



Imágenes [Repaso]

- HTML5 provee un “*ImageData*” .
- La imagen se recorre en *Ancho* y *Alto* (*for*, *for*)
- “*ImageData*” almacena los pixeles en un arreglo de 1d
- Los colores se almacenan en un *Array de enteros* y se pueden acceder como si fueran una matriz (**será?**)

- Para dibujar la Imagen en Pantalla: **ctx.drawImage(this, 0, 0);**
- Para convertir de matriz a arreglo?

indice = ...

*indice = (x + y * imageData.width) * 4;*

Acceder a los pixeles

- No puedo acceder a los pixeles de una Imagen directamente



- Entonces, para acceder a los pixeles de una imagen...

¡Primero tengo que dibujar la imagen!

```
ctx.drawImage(this, 0, 0);
```

No era esto?

Acceder a los pixeles

```
var imageData;
```

```
var imagen1 = new Image();  
imagen1.src = "image.jpg";
```

```
imagen1.onload = function() {
```

```
    ctx.drawImage(this, 0, 0, this.width, this.height);
```

```
    imageData = ctx.getImageData(0, 0, this.width, this.height);
```

```
    // Acceder a los pixeles ...
```

```
    ctx.putImageData(imageData, 0, 0);
```

✘ Uncaught DOMException: Failed to execute 'getImageData' on 'CanvasRenderingContext2D': The canvas has been tainted by cross-origin data. CargarImagen.html:38

IMPORTANTE: Por políticas de seguridad, Chrome **NO** permite acceder directamente a los datos de la imagen.

Acceder a pixeles [patch1]

```
// getting a hold of the file reference
```

```
let file = e.target.files[0];
```

```
// setting up the reader
```

```
let reader = new FileReader();
```

```
reader.readAsDataURL(file); // this is reading as data url
```

```
// here we tell the reader what to do when it's done reading...
```

```
reader.onload = readerEvent => {
```

```
    let content = readerEvent.target.result; // this is the content!
```

```
    let image = new Image();
```


Acceder a los pixeles [seguimos...]

```
// draw image on canvas
context.drawImage(this, 0, 0, imageScaledWidth, imageScaledHeight);

// get imageData from content of canvas
let imageData = context.getImageData(0, 0, imageScaledWidth, imageScaledHeight);

// modify imageData
for (let j = 0; j < imageData.height; j++) {
  for (let i = 0; i < imageData.width; i++) {
```

[Aquí hacer varias cosas] → filtros, etc

Acceder a los pixeles

- ¿Cómo accedo a cada pixel?
- ¿Cómo obtengo las componentes R, G y B?

```
function getRed(imageData, x, y) {  
    index = (x + y * imageData.width) * 4;  
    return imageData.data[index+0];  
}
```

```
function getGreen(imageData, x, y) {  
    index = (x + y * imageData.width) * 4;  
    return imageData.data[index+1];  
}
```

```
function getBlue(imageData, x, y) {  
    index = (x + y * imageData.width) * 4;  
    return imageData.data[index+2];  
}
```

Filtros

- Modifica el valor de un pixel dado una ecuación matemática
- Este valor puede ser **simplemente un coeficiente** (qué filtro sería?)



Grey

```
for (x=0;x++;x<image.width)
for (y=0;y++;y<image.height)
    r=getR(image[x,y]);
    g=getG(image[x,y]);
    b=getB(image[x,y]);

    grey=(r+g+b)/3;

    image[x,y].r=grey;
    image[x,y].g=grey;
    image[x,y].b=grey;
}
```

Tres valores, tengo que igualarlos....



$$\text{Gris} = (R + G + B) / 3$$

Sepia [conejito de indias]

https://www.google.com/search?q=sepia+image+processing&rlz=1C1CHZL_esAR696AR696&oq=sepia+image+processin&aqs=chrome.0.0j69i57j69i60l2.4850j0j4&sourceid=chrome&ie=UTF-8

