

2º DAW

DISEÑO DE INTERFACES WEB

Isaac Expósito López



iexposito103@educarex.es

Contenidos

- ❖ Git & GitHub (introducción breve)
- ❖ Diseño visual: Colores, tipografías...
- ❖ Multimedia
 - ❖ Imagen, audio y vídeo.
 - ❖ Animaciones.
- ❖ Maquetación tradicional
 - ❖ Responsive design
 - ❖ Media queries
 - ❖ Propiedades float y position
 - ❖ Layouts en columnas
- ❖ Maquetación moderna
 - ❖ Flex
 - ❖ Grid
 - ❖ Bootstrap
- ❖ Accesibilidad y usabilidad
- ❖ SEO. Posicionamiento Web

Material

Todo el material que necesitéis
os lo proporcionará Internet y yo

isaac-exposito.github.io



Classroom



Debéis tener email
de educarex activo



GitHub

C  DEPEN



Evaluación. RRAA (I)

1. Planifica la creación de una interfaz web valorando y aplicando **especificaciones de diseño**.
2. Crea **interfaces web** homogéneos definiendo y aplicando **estilos**.
3. **Prepara archivos multimedia** para la web, analizando sus características y manejando herramientas específicas.
4. **Integra contenido multimedia** en documentos web valorando su aportación y seleccionando adecuadamente los elementos interactivos.
5. Desarrolla interfaces web **accesibles**, analizando las pautas establecidas y aplicando técnicas de verificación.
6. Desarrolla interfaces web amigables analizando y aplicando las pautas de **usabilidad** establecidas.

Evaluación. RRAA (II)

RRAA	Porcentaje	Contenidos
1. Diseño visual	10 %	Colores, tipografías, contraste...
2. Creación de interfaces	60 %	Maquetación tradicional, responsive design, flex, grid, Bootstrap
3. Preparación multimedia	5 %	Formatos de imagen, audio y vídeo. Herramientas de manipulación. Animaciones
4. Integración multimedia	5 %	Integración de imagen, audio, vídeo y animaciones en documentos html
5. Accesibilidad	10 %	Accesibilidad
6. Usabilidad	10 %	Usabilidad

Evaluación

¡Hay que aprobar todos los RRAA!

EXÁMENES

- 1 o 2 por evaluación.
- ¿Cómo serán? Ni idea
- Recup. en enero y marzo
- No se estudia, se practica

PRÁCTICAS

- No obligatorias y no evaluables.

Uso de IAs

Debéis usarlas intensamente, pero usarlas bien



- Para resolver dudas
- Para mejorar tu propio código
- Para generar código que tú ya sabes hacer



- Para generar código que tú NO sabes hacer

Tip adicional

FP: Formación que demandan las empresas

➡ ¿Os enseñamos a hacer el trabajo lo más rápido posible?
Meh. Al menos ponemos plazos.

➡ ¿Os enseñamos a hacer el trabajo lo mejor posible?
Muy pocas veces. ¡Error nuestro!

Tip adicional

Dos ejemplos

Ejemplo 1:

FP de arquitectura.

Práctica: Diseño de un acueducto de piedra.

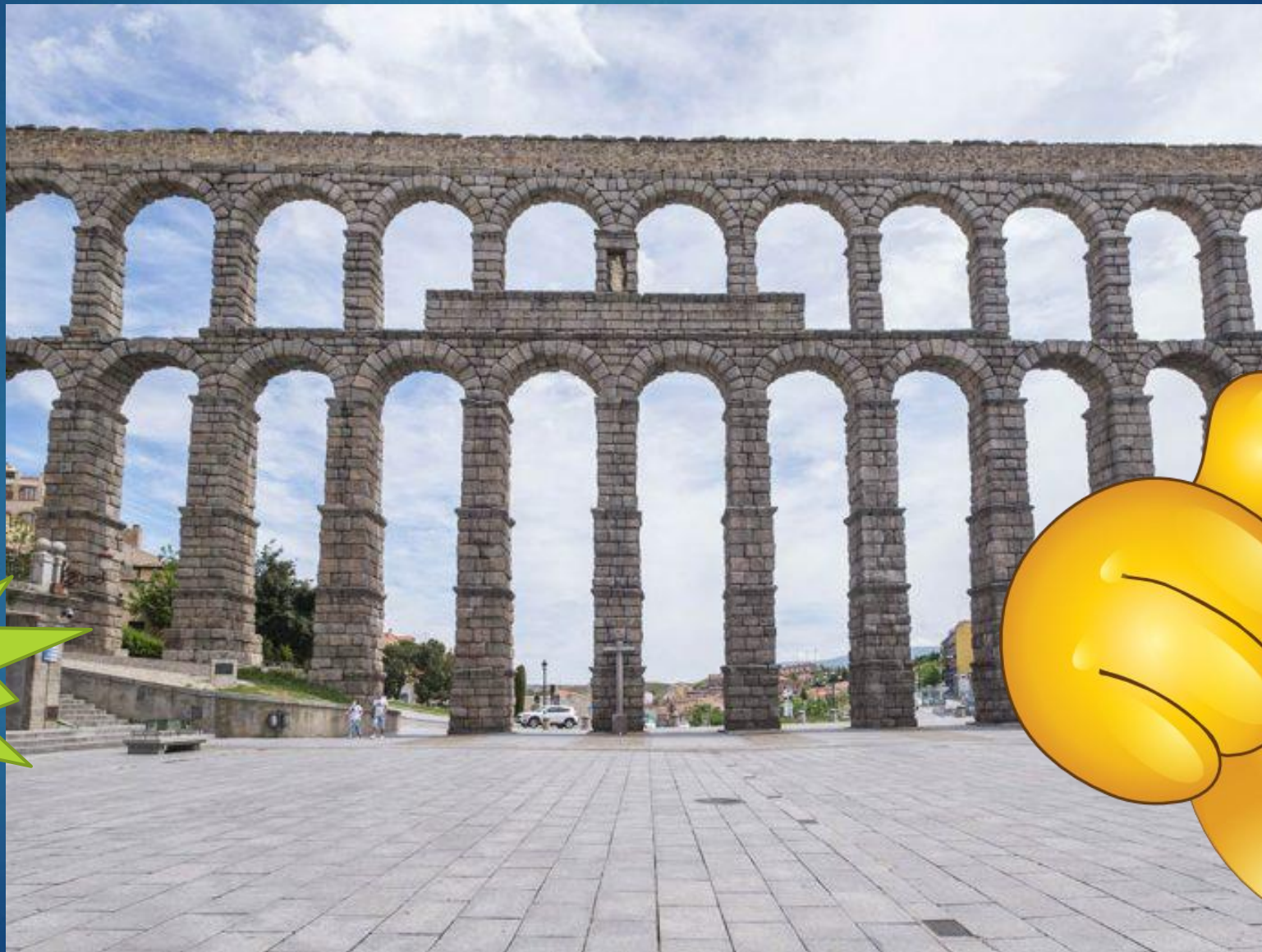
María y Juan entregan la práctica.

Tip adicional



María

10



Tip adicional



Juan

2



Tip adicional

Las cosas, además de funcionar,
deben estar bien hechas

Tip adicional

Ejemplo 2:

FP de informática.

Práctica: Diseño de un algoritmo para saber si un determinado elemento está en un array.

María y Juan entregan la práctica (de nuevo).


```
boolean encontrado = false;
for (int i = 0; i < array.length; i++) {
    if (array[i] == valorBuscado) {
        encontrado = true;
    }
}
```



Juan

```
boolean encontrado = false;
int i = 0;
while (!encontrado && i < array.length) {
    if (array[i] == valorBuscado) {
        encontrado = true;
    }
    i++;
}
```



María

¿Qué ocurrirá con un array de MIL MILLONES de elementos del 0 al 1 millón?

```
int min = 0;
int max = 1000;
int elementos = 1000000000;
int[] array = new int[elementos];
for (int i = 0; i < elementos; i++) {
    array[i] = (int) Math.floor(Math.random() * (max - min + 1) + min);
}
int valorBuscado = (int) Math.floor(Math.random() * (max - min + 1) + min);

long startTimeFor = System.nanoTime();
BuscarElemento.conFor(array, valorBuscado);
long elapsedTimeFor = System.nanoTime() - startTimeFor;
System.out.println("Tiempo de ejecución con bucle for: "
    + elapsedTimeFor / 1000 + " microsegundos");

long startTimeWhile = System.nanoTime();
BuscarElemento.conWhile(array, valorBuscado);
long elapsedTimeWhile = System.nanoTime() - startTimeWhile;
System.out.println("Tiempo de ejecución con bucle while: "
    + elapsedTimeWhile / 1000 + " microsegundos");
```



¿Qué ocurrirá con un array de MIL MILLONES de elementos del 0 al 1 millón?

```
compile:
```

```
run:
```

```
Tiempo de ejecución con bucle for: 3416 microsegundos
```

```
Tiempo de ejecución con bucle while: 41 microsegundos
```



```
boolean encontrado = false;
for (int i = 0; i < array.length; i++) {
    if (array[i] == valorBuscado) {
        encontrado = true;
    }
}
```

Funciona, pero **no está bien hecho**



Juan

```
boolean encontrado = false;
int i = 0;
while (!encontrado && i < array.length) {
    if (array[i] == valorBuscado) {
        encontrado = true;
    }
    i++;
}
```

Funciona y **está bien hecho**



María

¿Conocimientos previos?

- Git / GitHub
- HTML / CSS
- Flex, Grid
- *Responsive design*
- Bootstrap
- Accesibilidad/Usabilidad
- JavaScript

Sobre vosotros

- Nombre.
- Edad.
- ¿Tienes otros estudios?
- ¿Por qué DAW?
- Situación en el ciclo.
- ¿Seguirás estudiando?
- ¿Experiencia laboral?
- ¿Internet en casa?
- ¿Portátil?
- IDE favorito.