Documentación

Desarrollo de Interfaces

Programa: Workers con Foto

Autor: Cristina Fuster García

# **1. Documentación de usuario final:**

## **1.1. Guía de usuario**

Lo que vamos a encontrar en este sistema es la funcion de trabajar con workers, como crear nuevos numeros de trabajadores y contar cuanto tarda cada worker en recorrer la imagen y cambiarla de color a escalas grises.

Pasos para su uso

1. Colocar los archivos en una carpeta del proyecto metiendo el proyecto en la carpeta de htdocs junto con la imagen.
2. Abrir en el navegador el archivo, en la dirección de localhost.

## **1.2. FAQ (Preguntas frecuentes)**

¿Podemos cambiar el número de trabajadores?

Si entramos en el código podemos acceder al array numero de trabajadores y podemos poner el numero que queramos.

¿Hay que tener en cuenta los núcleos de la CPU?

SI, siempre es recomendable saber de cuántos núcleos dispones para que el ordenador no se quede bloqueado en los procesos.

# **2. Documentación técnica para desarrolladores**

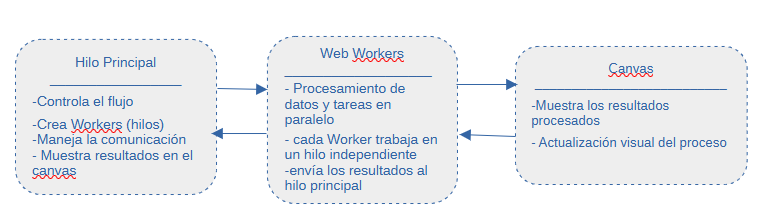
## **2.1. Introducción al software**

Es un sistema para comprobar el rendimiento del ordenador, creamos varios workers que son los responsables de dividir la faena del ordenador y no sovrecargarlo, en este ejemplo vamos a ver como podemos crear el numero de workers y la comunicacion entres ellos por onmassage y postmassage.

Tecnologías Utilizadas

* html
* JS (web sockets)

## **2.2. Arquitectura del sistema**



Hilo principal cargar el archivo html ejecuta el script con el número detectado de los núcleos en la cpu y luego crea los workers según la capacidad del núcleo

El web Worker es el que trabaja con los hilos procesando los datos y tareas en paralelo.

En el canvas es donde se va a ver el resultado de la imagen despues de realizar las tareas.

## **2.2. Estructura del código**

Carpeta /

* archivo.html
* archivo.js
* imagen

Funciones de código

* En el archivo html tenemos un elemento canvas que contiene la imagen. Y también tenemos el script de JavaScript donde está la lógica.
* donde vamos a crear los worker.
* Hacemos la reparticion de la imagen a cada worker y se lo enviamos al js mediante el postMessage.
* Implementamos también un contador y localizamos a los workers por su id
* En Javascript tenemos la función de onmessage que se ejecutará una vez reciba el resultado.

Tecnologias:

* html
* js

# **3. Guías de instalación y despliegue**

## **3.1. Requisitos previos**

Tener instalado el xampp, y un navegador compatible.

## **3.2. Instrucciones de instalación**

1.Descagar los archivos del proyecto

* Instalar el servidor

instalar el xampp una vez esté instalado abriremos el panel de control y encender el apache.

* Tendremos que descargar el proyecto o crearlo en la carpeta de htdocs
* Abrir el archivo en el navegador en la dirección de localhost : http:/localhost/nombre del archivo.

# **4. Mantenimiento y soporte**

## 

## **4.1. Manejo de errores**

Que la imagen no aparezca, significa que la ruta no está correctamente cargada.

## **4.2. Registro de cambios (changelog)**

verion 4: estuvimos trabajando la version pero sin workers de intermediario, solo convertimos la foto c¡de color en gris con el Math.roud-

versión 5: capturamos la fecha y la hora con los milisegundos con date() y getTime()

versión 6: empecemos a trabajar con los workers en este caso en concreto solo teníamos a uno.

**4.3. Soporte y contacto**