

Documentación de Desarrollo de Software

Plataforma de software: Mapa RNI

Proyecto: Visualización de los niveles de radiación no ionizante en un mapa digital de la ciudad de San José de Cúcuta

V. 1.0.0

1. Estructura

El proyecto contiene la siguiente estructura de carpetas en su código fuente:

- **CSS:** Carpeta con los estilos que aportan el diseño del sitio. Fundamentalmente esta carpeta contiene bootstrap.min.css (librería utilizada para la correcta visualización en todos los navegadores) y styles.css (los estilos propios de la aplicación). styles.css también se encuentra como styles.min.css, una versión con el mismo contenido, pero de forma comprimida para su rápida descarga a través de internet.
- **Documentos:** Carpeta con diferentes documentos PDF de apoyo que se muestran desde el menú superior de la aplicación.
- **Excel:** Librería para la conversión de los datos de la base de datos a un formato excel que el usuario puede descargar.
- **Iconos:** Iconografía gráfica utilizada en la aplicación (logos y paletas de colores).
- **Img:** Imágenes mostradas en la aplicación (equipos, campaña, etc.).
- **Js:** Código javascript de la aplicación. Fundamentalmente contiene bootstrap (librería de diseño), jquery (librería de utilidades) y main.js, el núcleo central de la aplicación.
- **Upload:** Carpeta donde se almacenan temporalmente los archivos CSV subidos para su posterior importación a la base de datos.

Del mismo modo, algunos archivos fundamentales de la aplicación se encuentran directamente en la raíz de la aplicación. Estos son archivos HTML que despliegan información en el navegador, y archivos PHP que realizan operaciones en el servidor antes de mostrar al usuario datos en pantalla.

2. Aplicación PHP

Como se ha definido en la sección arquitectural del documento de diseño, la aplicación se plantea con dos componentes, uno de cliente y uno de servidor. Toda la aplicación de servidor es construida utilizando PHP como lenguaje de programación. A continuación, se detalla el funcionamiento de los componentes PHP que componen la aplicación:

Componente	Función
borrar.php	Este componente recibe el número identificador de una medición como parámetro. Busca en la base de datos la medición que contiene ese identificador, la elimina, y retorna al usuario a la vista de base de datos.
campana.php	Despliega información sobre la campaña de medición y las herramientas utilizadas en el proceso. Muestra información e imágenes relacionadas, así como acceso directo a documentos con información de las herramientas.
cierre_sesión.php	Termina la sesión de un usuario, lo que anula su acceso a la base de datos y lo dirige a la página principal con el mapa.
comprueba_login.php	Recibe como parámetros los datos que el usuario ha provisto al intentar iniciar sesión: su nombre de usuario y contraseña. En este componente se encripta la contraseña y posteriormente se verifica que los datos coincidan con los almacenados en la base de datos. Si coinciden, el usuario es redirigido a la vista de base de datos. En caso contrario su solicitud de inicio de sesión es rechazada, y vuelve a la vista original.
conexión.php	Módulo encargado de la conexión con la base de datos. En lugar de proveer los datos de conexión cada vez que sean necesarios, estos se mantienen aislados en este archivo, el cual es importado por cualquier otro que necesite acceder a la base de datos.
editar.php	Recibe como parámetros todos los datos medidos para un punto específico y el usuario puede cambiarlos a través de un sencillo formulario. Una vez estos datos han sido completamente editados, almacena la información en la base de datos.
index.php	Componente que despliega toda la información de la base de datos a los usuarios autenticados. En esta sección se obtienen todos los datos desde la base de datos y se muestran en una tabla que contiene paginación cada 10 valores. Desde esta tabla puede editarse o eliminarse cualquier medición, y por tanto solo es accesible para aquellos usuarios que inicien sesión correctamente con cuenta y contraseña.
reporte.php	Este componente toma como base la librería PHPExcel y con ella, toma los datos almacenados en la base de datos y los convierte en un archivo de Excel

(.xlsx) que el usuario puede descargar y almacenar en su dispositivo. Este componente solo está disponible para usuarios que han iniciado sesión con su cuenta y contraseña.

subir_archivo.php	Este componente se encarga de tomar archivos CSV subidos por usuarios autenticados, guardarlos temporalmente y tomar de ellos la información para posteriormente almacenarla en la base de datos.
XML.php	Componente fundamental para la visualización del mapa con la información medida y las interpolaciones: Este componente toma la información de la base de datos y la convierte en un archivo XML que es enviado al navegador, y con base a él se despliegan todas las visualizaciones.

3. Puntos de entrada HTML

La aplicación en el servidor se construye con PHP, así como en el cliente estará basada con código JavaScript. Sin embargo, también como parte del cliente se encuentran archivos HTML que son el punto de entrada a través del cual el usuario interactúa con la aplicación. Estos puntos de entrada son:

Componente	Función
index.html	Página principal de la aplicación. Es el archivo que se muestra al usuario al ingresar al link. Este componente contiene el mapa, eje central de la aplicación, haciendo uso de los demás componentes externos, de cliente y de servidor.
navbar.html	La aplicación contiene diferentes vistas, pero todas comparten un mismo menú superior. Por tanto, en vez de repetir este menú en cada uno de los archivos que lo soliciten, se opta por almacenarlo en el componente navbar.html e importarlo en cada sitio que sea requerido. Este navbar contiene información de referencia de diferentes entidades reguladoras sobre la radiación no ionizante.
footer.html	Al igual que con el navbar, todas las vistas de la aplicación poseen al final un pie de página con información institucional y de contacto. Para no reescribir el mismo contenido varias veces, se opta por almacenar esta información en el componente footer.html e importarlo cada vez que sea requerido.

4. Aplicación JavaScript

La parte más compleja de la aplicación se encuentra en el código que ejecuta en el navegador: Allí se despliega el mapa, se muestra la información y se realizan todas las interpolaciones. Todo ello ocurre a través de un único archivo main.js. Si bien podría separarse en varios componentes como ocurre con la aplicación PHP, esto conlleva mayores demoras en la carga del sitio, especialmente bajo condiciones inestables de conectividad. Por tanto, se opta por mantener un único archivo con todos los componentes en él.

A continuación, se detallan cada uno de los métodos que contiene el archivo main.js, su descripción, sus parámetros y la respuesta que retorna, si aplica.

Los nombres de las funciones en el código son expresados en idioma inglés, para mantener la coherencia con el lenguaje de programación. Por tanto, a continuación, los títulos de cada función y sus parámetros se encuentran en inglés, aunque las descripciones se encuentren en español.

Función:	showPalette	
Descripción:	Muestra paleta de colores sobrepuesta en el mapa	
Parámetros:	controlDiv	Div sobre el cual se ubica la paleta.
	map	Mapa sobre el cual se ubica la paleta.
Retorno:	No aplica.	

Función: `showOpacityButton`

Descripción: Muestra botón para cambiar opacidad sobrepuesto en el mapa.

Parámetros: `controlDiv` Div sobre el cual se ubica el botón.
`map` Mapa sobre el cual se ubica el botón.

Retorno: No aplica.

Función: `bound`

Descripción: Verifica que un valor (`value`) se encuentre en el rango [`optMin`, `optMax`]. En caso de no estar en ese rango, retorna el valor más cercano que sí esté en dicho rango. Usado para establecer límites en las coordenadas.

Parámetros: `value` Valor a verificar en el rango.
`optMin` Límite inferior del rango.
`optMax` Límite superior del rango

Retorno: Si el valor está en el rango, retorna el mismo valor. En caso contrario retorna el valor más cercano dentro del rango.

Función: `degreesToRadians`

Descripción: Convierte un valor en grados sexagesimales a radianes.

Parámetros: `deg` Grados sexagesimales a convertir.

Retorno: Valor convertido en radianes.

Función: radiansToDegrees

Descripción: Convierte un valor en radianes a grados sexagesimales.

Parámetros: rad Radianes a convertir.

Retorno: Valor convertido en grados sexagesimales.

Función: MercatorProjection

Descripción: Genera una proyección Mercator sobre el mapa para conservar el mapa de calor al hacer zoom sobre él.

Parámetros: Ninguno.

Retorno: No aplica.

Función: MercatorProjection.prototype.fromLatLngToPoint

Descripción: Convierte las coordenadas de longitud/latitud a un punto ubicable en el plano.

Parámetros: latLng Objeto lanLng con las coordenadas.

optPoint Punto de origen.

Retorno: Punto convertido.

Función: MercatorProjection.prototype.fromPointToLatLng

Descripción: Convierte un punto ubicable en el plano en coordenadas geográficas.

Parámetros: point Punto ubicable en el plano.

Retorno: Coordenadas del punto en el formato LatLng de Google Maps.

Función: `getNewRadius`

Descripción: Obtiene el nuevo radio del mapa al realizar zoom sobre él.

Parámetros: Ninguno

Retorno: Nuevo radio del mapa visible.

Función: `getPoints`

Descripción: Obtiene los puntos desde la base de datos, a través de un archivo XML.

Parámetros: Ninguno

Retorno: Arreglo con todos los datos obtenidos.

Función: `loadData`

Descripción: Despliega la información obtenida desde la base de datos, en el mapa.

Parámetros: `datos` Datos obtenidos desde la base de datos.

Retorno: No aplica.

Función: `initMap`

Descripción: Inicializa el mapa con la información por defecto.

Parámetros: Ninguno

Retorno: No aplica.

Función: `changeGradient`

Descripción: Establece el radiante para el mapa de calor.

Parámetros: Ninguno

Retorno: No aplica.

Función: `getLocation`

Descripción: Obtiene la posición a través de la geolocalización del navegador.

Parámetros: Ninguno

Retorno: No aplica.

Función: `interpolate`

Descripción: Toma una posición dada y estima su valor utilizando interpolación teniendo como referencia los puntos a su alrededor.

Parámetros: `lat` Latitud del punto a interpolar.

`lng` Longitud del punto a interpolar.

`imageMarker` Ruta de la imagen del marcador a mostrar.

`setCenter` True si el mapa debe centrarse en el punto elegido. False en caso contrario.

`onlyCampus` Si esta variable es true, se mostrará mensaje de error si el usuario está fuera del campus.

`markersToSet` Marcadores a tener en cuenta.

Retorno: No aplica.

Función: showSelectedPosition

Descripción: Muestra la ubicación del marcador añadido con la información necesaria. Para esto realiza interpolación de valores.

Parámetros: Ninguno

Retorno: No aplica.

Función: showMyPosition

Descripción: Muestra en pantalla la posición actual del usuario y su medición estimada realizando interpolación con los puntos ubicados alrededor.

Parámetros: Position Ubicación actual del usuario

Retorno: No aplica.

Función: showMeasuredPoints

Descripción: Muestra en el mapa un marcador por cada punto almacenado en la base de datos.

Parámetros: Ninguno

Retorno: No aplica.

Función: addMarkers

Descripción: Añade un marcador en el mapa.

Parámetros: pos Coordenadas (longitud, latitud) del punto.

image Imagen del marcador.

`imageMarker` Ruta de la imagen del marcador a mostrar.

`infoWindow` Objeto de información.

`infoWindowContent` Contenido del objeto de información.

`markersToSet` Arreglo de marcadores en el cual se añadirá

`showDefault` True si la información se muestra por defecto. False si solo se muestra la imagen del marcador.

Retorno: No aplica.

Función: `setMapOnAll`

Descripción: Activa o desactiva todos los marcadores de un tipo específico.

Parámetros: `markersToSet` Marcadores a activar o desactivar.

`map` Mapa actual.

Retorno: No aplica.

Función: `showOrHide`

Descripción: Muestra u oculta el mapa de calor sobrepuesto sobre el mapa.

Parámetros: Ninguno

Retorno: No aplica.

Función:	downloadUrl	
Descripción:	Descarga información desde el servidor.	
Parámetros:	url	Link desde el cual se descarga la información.
	callback	Acción a realizar con la información una vez descargada
Retorno:	No aplica.	

Función:	handleLocationError	
Descripción:	Informa al usuario sobre posibles errores en la geolocalización.	
Parámetros:	browserHasGeolocalization	False si el navegador no posee geolocalización. True en caso contrario.
	infoWindow	Objeto de información donde se avisará al usuario sobre posibles errores.
	pos	Ubicación donde se mostrará el error.
Retorno:	No aplica.	

Función:	setOpacity	
Descripción:	Añade o elimina opacidad al mapa	
Parámetros:	Ninguno	
Retorno:	No aplica.	
