# Procedimiento

## -Google Sheet como base de datos

Las hojas de Google evolucionan día a día ganando cada vez más poder convirtiéndose hoy en día en una de las mejores aplicaciones de foja de cálculo.

Google sheets tiene aplicaciones como Apps scrit, Form triggers, Google sheers API, Timed triggers, que lo convierte en una herramienta mas completa si lo comparamos con Excel.

Hablaremos de las características de Google sheet, las limitaciones que presenta esta aplicación, las soluciones alternativas que pueden utilizarse ante los problemas que se presentan y como usar Google sheet como base de datos.

Pataforma App scripr para desarrollarores.

Google contiene muchas aplicaciones integradas que pueden interactuar con Google sheet App script como MailApp , DriveApp, DocumentApp, SlidesApp, CalendarApp etc.

Hay muchas integraciones disponibles para trabajar con Google sheets pero la mayoría de las funciones no se suelen usar debido a que muchos de los usuarios y desarrolladores ni siquiera saben todo lo que Google App Script puede proporcionarles.

Uso y limitaciones de la plataforma de App script.

Como hemos dicho anteriormente, App script tiene muchas integraciones disponibles dentro de todo en entorno de Google. App Script es muy útil cuando deseamos compartir y modificar datos entre las aplicaciones de Google. Como por ejemplo.

-Podríamos responder automáticamente a algún usuario cuando envíe un formulario de Google usando MailApp.

-Envío periódico a usuarios que se encuentren en una hoja de cálculo de Google con MailApp y Timed triggers.

-Sincronizar eventos entre hojas de cálculo de Google y Google Calendar.

Estos son solo algunos de los ejemplos entre todas las posibilidades que nos ofrecen las aplicaciones de Google y con la que podríamos montar una base de datos.

## -Revisar alternativas para móvil

Los desafíos a los que se enfrenta la enseñanza hacen que se busquen continuamente nuevos métodos para hacer frente a estos problemas.

Con la intención de facilitar la programación de aplicaciones móviles surge App Inventor mediante un lenguaje visual basado en bloques.

App Inventor para Android, es un entorno visual de programación en bloques, que posibilita el desarrollo de aplicaciones para dispositivos móviles Android de forma bastante sencilla sobre todo si es comparado con los lenguajes de programación tradicionales.

### El entorno App Inventor

App Inventor es un lenguaje visual de programación para la creación de aplicaciones para dispositivos móviles, desarrollada conjuntamente por Google y MIT (Instituto Tecnológico de Massachusetts). App Inventor esta basado en un entorno de desarrollo compuesto de bloques, sin embargo, la diferencia fundamental radica en permitir a los usuarios construir aplicaciones que disponen de servicios basados en la web, lectura de códigos de barras, interacción con redes sociales, interacción con sensores de orientación y geolocalización.

La creación de aplicaciones con App Inventor es bastante intuitiva y no requiere conocimientos previos avanzados en programación. Además, las aplicaciones creadas se pueden utilizar en cualquier dispositivo con plataforma Android. App Inventor esta basado en componentes los cuales el usuario los manipula mediante la interacción con la aplicación.

La construcción de una aplicación en esta plataforma se realiza a través de dos espacios:

App Inventor Designer y Blocks Editor. App Inventor Designer puede ejecutarse desde cualquier navegador, gracias a el podemos crear visualmente el interfaz de la aplicación simplemente haciendo clic y arrestando componentes como botones, cuadros de texto, figuras, sonidos o animaciones que se encuentran en Palette.

En la ventana de Bock Editor, se encuentran bloques que podemos conectar, estos bloques también se pueden añadir a la pantalla haciendo clic y a arrastrando hacia el espacio de trabajo, con ellos podemos controlar de qué manera se comportaran los componentes que previamente se han definido en la ventana de App Inventor Designer.

La aplicación nos permite realizar pruebas mediante un emulador o instalar la aplicación y ejecutarla directamente desde un dispositivo Android (smartphone o tablet).

Ventajas de utilizar App inventor respecto a otros entornos.

Utilizar App inventor trae consigo ventajas respecto a otras plataformas que podrían dar resultados similares al que se busca en este trabajo.

La ventaja principal de crear una App nativa (es una aplicación desarrollada para smartphones en el lenguaje nativo del propio terminal) respecto a una Web App es que la App nativa una vez descargada e instalada en el smartphone estará siempre a disposición del usuario sin necesidad de acceder a internet cada vez que quiera acceder a ella.

Otras ventajas de que la aplicación esté instalada en el dispositivo móvil son la posibilidad de acceder a elementos como la cámara o la ubicación además de ser más rápida que una web App.

Las ventajas de utilizar App inventor respecto a otras plataformas de programación de App nativas está en que el lenguaje de programación de App inventor consiste en un entorno visual de programación con bloques. Esto hace más sencillo la creación de la aplicación ya que no es necesario saber otros lenguajes de programación como Java, JavaScript, HTML5…

Estas ventajas hacen que utilizar App inventor para realizar la aplicación sea la mejor opción.

## -Modelo de base de datos

La base de datos está compuesta por 4 tablas la primera tabla se utiliza para el registro de los usuarios. La segunda tabla está compuesta por el Id de los usuarios, el Id de los fallos que puede tener la máquina, el id de la máquina a la que corresponda el fallo y otros datos que ayuden a resolver el problema. Por último, en las tablas restantes irán los fallos junto con su correspondiente identificador y en la última tabla estarán los identificadores de las maquinas asociadas a los datos correspondientes para su localización.

La tabla de registro está compuesta por la columna id que irá completando automáticamente al recibir el dato de la columna usuario y correo. Cada usuario solo podrá aparecer una vez en la tabla, de este modo únicamente tendrá un valor de id.

La tabla de incidencias recibirá los datos introducidos por el usuario correspondientes a la incidencia, estos datos se relacionarán con la tabla de registro mediante el valor de id del usuario al que irán asociados también a la incidencia en la segunda tabla.

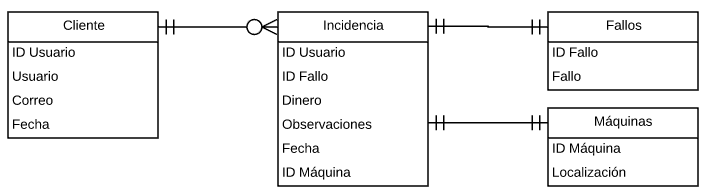
|  |  |  |  |
| --- | --- | --- | --- |
| **Registro** | | | |
| **Id Usuario** | **Usuario** | **Correo** | **Fecha** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Incidencias** | | | | | |
| **ID Usuario** | **ID Fallo** | **Dinero** | **Comentario** | **ID Máquina** | **Fecha** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Fallos** | |
| **ID Fallos** | **Fallo** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Maquinas** | |
| **ID Máquina** | **Localización** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

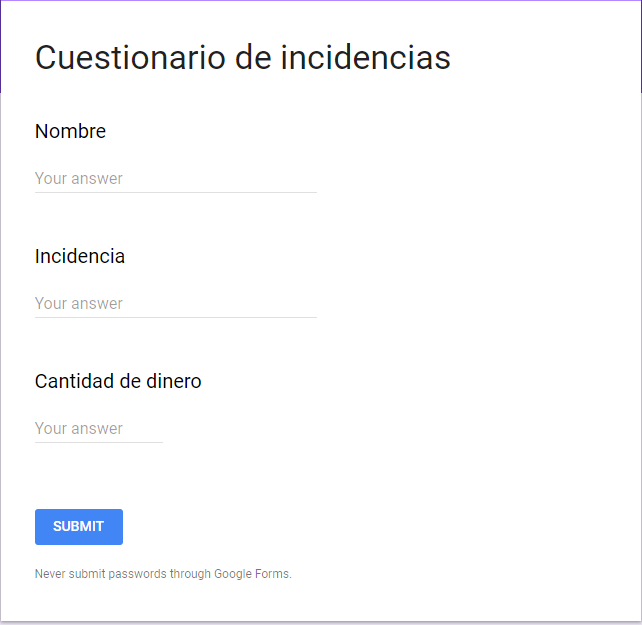
De este modo para un id solo puede existir un único usuario, pero puede haber varias filas en la tabla 2 asociadas a este id.



## -Aplicación dos formularios: identificación (solo una vez) e incidencias.

Para recibir las incidencias a la base de datos es necesario disponer de alguna plataforma en la que poder publicar las preguntas que está buscando el cliente afectado para que este pueda responder con su problema. Para esto Google Sheets ofrece la posibilidad de crear formularios y almacenar las respuestas a cada pregunta en columnas.

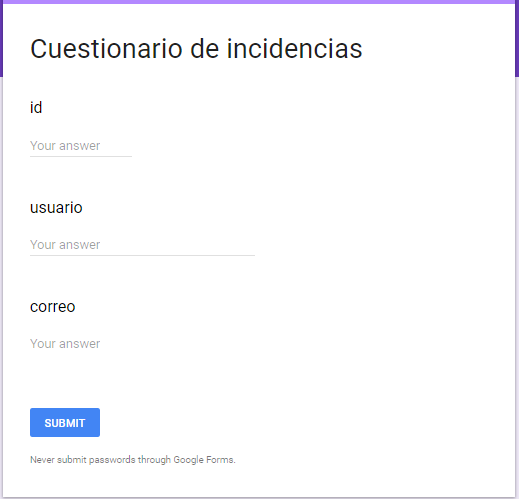
Podríamos crear un cuestionario con los apartados para nombre, incidencia, cantidad de dinero por si ese fuera el problema y quedaría de la siguiente forma.



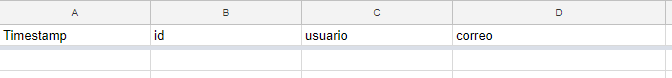
Pero este resultado es bastante primitivo aparte de que lo más correcto sería que en la base de datos estuvieran en tablas separadas los datos del usuario y las incidencias.

Por eso crearemos dos formularios, uno para almacenar los datos del usuario, y otro para registrar las incidencias. El objetivo es subir estos dos cuestionarios de alguna manera a la aplicación para almacenar los datos.

El cuestionario destinado para almacenar tendrá las secciones para almacenar: id, usuario y correo. Quedando de la siguiente manera.



El id será un numero único para cada usuario y será asignado por la base de datos, el usuario será el nombre con el que el cliente afectado quiera que le nombremos y el correo electrónico será necesario para informarle del estado en el que se encuentra su incidencia.



Como vemos se ha creado una columna para cada sesión del cuestionario, adema, podemos observar que aparte de crearse estas tres columnas también se ha creado una cuarta con la fecha en el que se envía el formulario. Ya tenemos creada la tabla con los campos necesarios para el registro en la App.

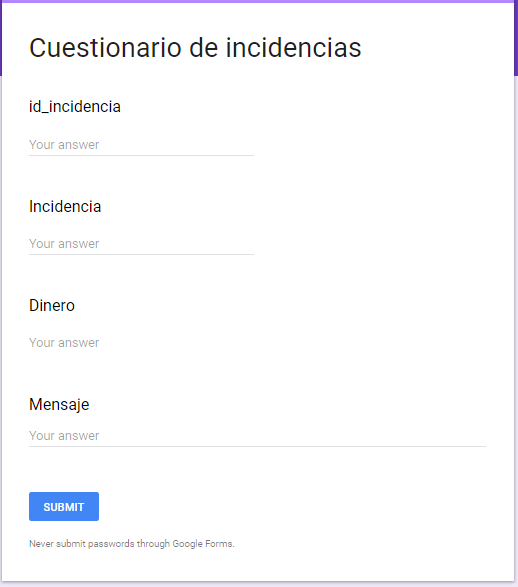
El cuestionario para almacenar las incidencias contendrá los siguientes campos:

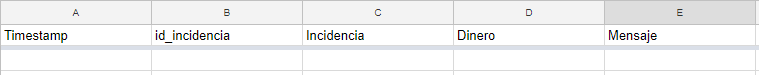
-Id\_incidencia, al igual que para el usuario cada incidencia tendrá un id único e irrepetible con el cual se hará referencia al conjunto de la incidencia cuando sea necesario.

-Incidencia, en este espacio el usuario podrá detallar el problema que ha tenido con la máquina.

-Dinero, cuando el contratiempo se deba a una devolución incorrecta de la maquina al cliente el usuario introducirá en este lugar la cantidad no devuelta por la máquina.

-Además, aparecerá otra cuestión llamada mensaje en la que la persona que esté rellenando el formulario podrá añadir más detalles de la incidencia si fuera necesario.



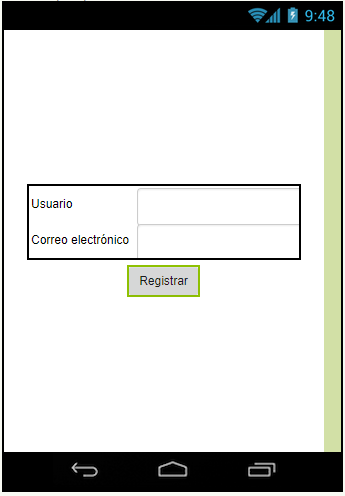


Al igual que antes podemos observar que aparte de crearse los encabezados de las columnas que hemos incluido en el formulario también se ha creado una columna con la fecha en la que se envía la incidencia.

Este dato resulta de gran interés ya que si conocemos la fecha en la que se registra la incidencia podremos clasificarla por tiempo o calcular la duración en resolver el problema.

Como dijimos en un punto anterior, utilizaremos App Inventor para crear la aplicación, para ello accedemos a la página web y creamos un nuevo proyecto.

Para crear el formulario de registro de usuario desde el apartado de diseño necesitaremos arrastrar a la pantalla dos cajas de texto, dos etiquetas de texto para identificar cada caja, un botón para enviar los datos del formulario y también arrastraremos la opción web que se encuentra en el apartado de conectividad, este es un componente no visible que permite gestionar solicitudes HTTP como GET, POST, PUT y DELETE una vez hecho quedara de la siguiente forma.

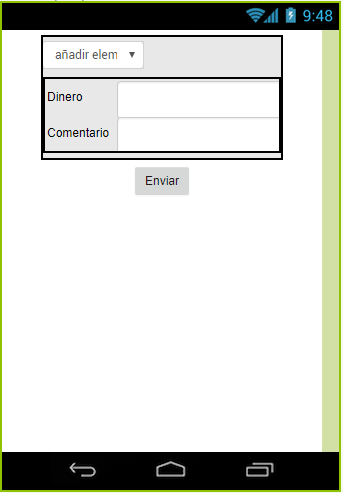


Ahora crearemos otra pantalla en la que construiremos otro formulario, este será par a introducir los datos referentes a la incidencia.

Añadiremos a la pantalla un desplegable en el que introduciremos las incidencias más comunes (no me ha devuelto el cambio, no hay agua, no hay café, no hay vaso, no hay azúcar, el líquido está frío, maquina no operativa y cantidad insuficiente) y la opción otro por si no estuviera entre estas.

Agregaremos también a la pantalla dos cajas de texto, una para introducir la cantidad económica que no ha devuelto la maquina si este fuera el problema y la otra para introducir la incidencia si la opción fuera otra a las que aparecen en la lista. Estas cajas de texto al depender de la selección de la incidencia de punto de partida serán invisibles y se verán o no en función de la selección del usuario.

También al igual que en el formulario de registro añadiremos la opción web de conectividad y un botón para enviar el formulario.

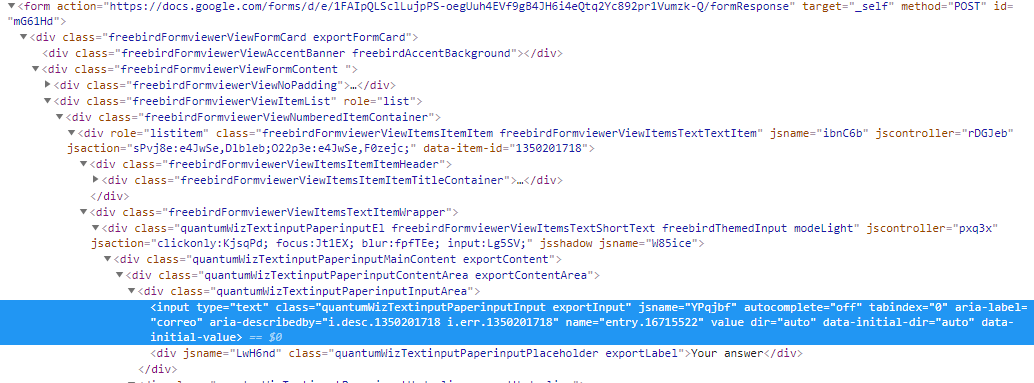


Desde la aplicación enviaremos los datos a la base de datos mediante un post, es decir enviaremos los datos desde un enlace.

Para ello lo primero que haremos será ir al formulario en vivo y mirar el código fuente del formulario. La URL de publicación se encuentra en la etiqueta acción de formulario y los elementos de la lista.



También hay que encontrar los elementos de la lista, buscaremos la etiqueta de nombre para encontrar un valor como “entry.16715522”.

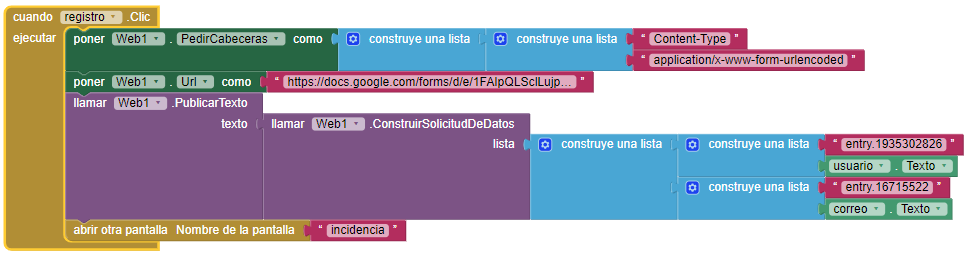


Esto lo tendremos que hacer tanto en el formulario de registro como en el de envío de incidencias.

Después iremos a la hoja de cálculo y seleccionaremos Archivo – Compartir, hacemos clic en “avanzado” y cambiamos el acceso de privado a cualquiera con enlace y cambiamos el tipo de acceso de “puede ver” que viene por defecto a “se puede editar” y hacemos clic en guardar.

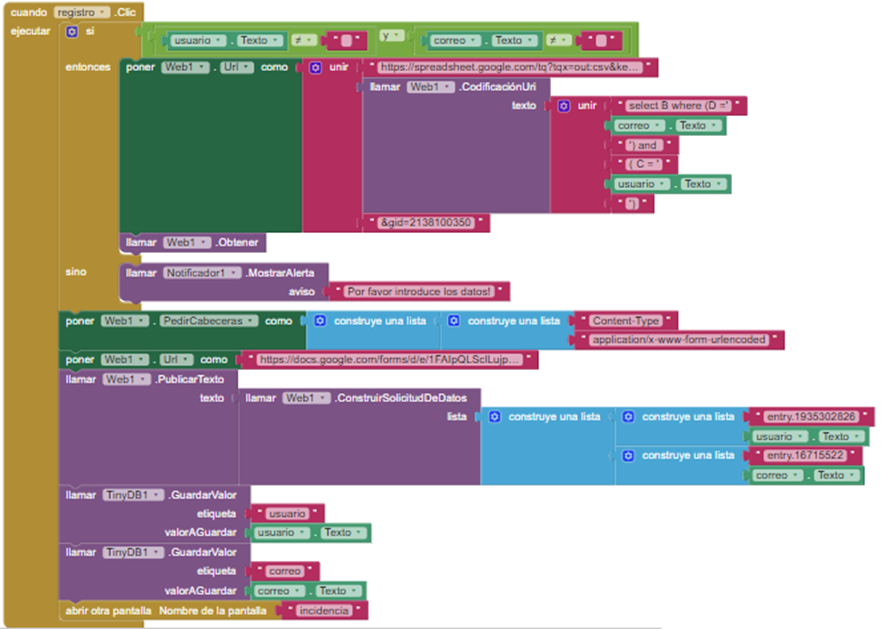
Ahora seleccionamos archivo – publicar en la web en la siguiente pantalla hacemos clic en publicar y luego en aceptar y cerrar.

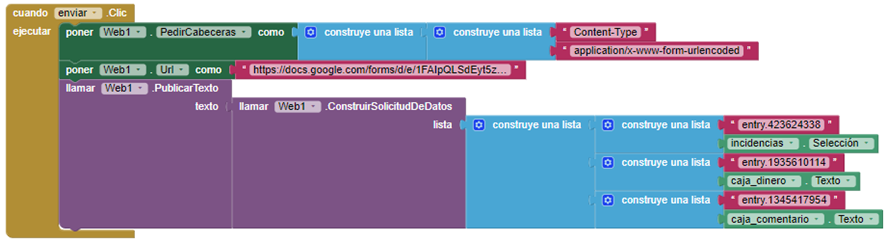
Hechos esto ahora construiremos los bloques. Realizaremos el envío de los datos de registro del usuario mediante un post utilizando el enlace obtenido del código fuente del formulario y los nombres de los elementos que pusimos en el de la siguiente forma.



Los datos referentes a las incidencias los enviaremos de la misma forma que los de registro de usuario, obtendremos el enlace del formulario y el nombre de los elementos que queremos conseguir.

En este formulario sin embargo veremos las cajas de texto de dinero y de comentarios en función de la selección como podemos ver en la imagen.





## -Relación usuario id

Anterior mente tanto en la tabla de registro de usuarios como en la de incidencias hemos creado un campo identificado como id. Este campo será muy útil para poder identificar cada incidencia o cada usuario por un número que será único en cada caso.

Este campo queremos que se cree automáticamente, para ello crearemos un script con un programa que nos realizará este trabajo.

Para ello desde la hoja de cálculo iremos a herramientas, script editor y crearemos un nuevo script.



Este programa lee el último dato de la columna de id y le suma +1. Este proceso se hará automáticamente cada vez que se envíe un formulario de usuario o de incidencias.

De este modo ya tenemos automatizada la creación del id para cada incidencia o usuario.

-Usuario recordado

Para que el uso de la aplicación sea lo más cómodo posible para el usuario lo ideal es que una vez registrado, la aplicación tiene que ser capaz de recordar sus datos.

Las aplicaciones creadas con App Inventor se inicializan cada vez que son ejecutadas. Esto quiere decir que si una aplicación fija el valor de una determinada variable y el usuario cierra la aplicación el valor de esta variable no será recordado la próxima vez que se inicialice la aplicación.

Para ello App Inventor dispone de TinyDB, este es un componente invisible que sirve para almacenar datos para la aplicación.

Los datos almacenados en TinyDB estarán disponibles cada vez que se abra la aplicación.

Los datos se guardan mediante valores con etiquetas. Para guardar un elemento hay que especificar la etiqueta con la que quiere guardarse. Luego para recuperar el valor solo hay que especificar la etiqueta con la que hemos guardado ese dato.

Para añadir este componente simplemente lo arrastramos desde los componentes de almacenamiento hasta nuestra pantalla, una vez añadido el componente nos vamos a los bloques donde lo configuraremos.

Añadiremos dos bloques de TinyDB en los cuales el valor a guardar será el texto escrito en los campos de texto de usuario y correo con las etiquetas usuario y correo que serán los nombre con los que se identificarán estas variables

Cuando el usuario haga clic en el botón de registrar automáticamente los valores de usuario y correo quedaran guardados en el dispositivo.

## -Crear lista de incidencias

El sistema de automatización de incidencias que crearemos estará compuesto también de una página web en la que se podrán ver los resultados de la aplicación.

Se podrán ver los datos de forma sencilla como incidencias y estadísticas que facilitarán la resolución de los problemas ocasionados por las máquinas de café.

La página web mostrará los datos de las tablas que se encuentran en Google Sheet, dicho de otro modo, la intención es mostrar las mismas tablas que tenemos en Google Sheet pero de una forma más simplificada y atractiva.

Crearemos unos archivos para almacenar el código que será necesario para la página web, estos archivos los guardaremos en un repositorio.

GitHub es una plataforma de desarrollo colaborativo de software para guardar proyectos utilizando el sistema de versiones Git.

|  |
| --- |
| No sé si debo mostrar los archivos la página web o que |

Para crear la página web con estos archivos utilizaremos una aplicación llamada RawGit, esta aplicación actúa como un servidor de almacenamiento en caché para archivos que estén subidos a GitHub. Es básicamente una aplicación de internet que sirve archivos de GitHub a CDN externos. RawGit reenvía todas las solicitudes de usuario a GitHub y luego almacena en cache las respuestas a su navegador.

Para obtener nuestra página web únicamente tenemos que obtener la dirección del archivo en GitHub, ir a la página de RawGit pegar el enlace en el apartado de archivos en bruto y automáticamente se creará una URL que será la dirección de la web.

El aspecto de la página web que hemos creado es siguiente, en la parte superior hay un menú en el que aparecen las opciones incidencias, estadísticas y mantenimiento para poder movernos fácilmente.

|  |
| --- |
| Aspecto de la página web sin tablas |

Tal como hemos visto en la página hay una sección para incidencias, en ellas se irán publicando tablas de incidencias con diversos motivos. Uno de estos gráficos que se publicarán será el de las ultimas incidencias. Pero publicar los gráficos tal cual no es estético, ni ofrece una vista cómoda de los datos, por ello hará que solucionar este problema.

La web nos proporciona infinidad de soluciones con las que resolver este problema, entre todas las posibilidades que hay en este caso se a optado por utilizar Awesome Table.

Awesome table es una aplicación capaz de convertir una información simple y aburrida de una hoja de datos de Google Sheet en una vista atractiva.

Awesome table se integra perfectamente en el entorno de Google, la única condición para utilizar la aplicación web es tener una cuenta de Google.

Con esta aplicación es posible dar un determinado formato a los datos para que se vean de la manera que nosotros queramos, además estos datos se pueden ordenar forma sencilla mediante filtros interactivos.

Aparte de todo esto, esta aplicación web nos permite que incrustemos en cualquier página web el resultado que hemos obtenido.

Accedemos a la página web de Awesome table, ingresamos nuestro usuario de Google, hacemos clic en crear una nueva vista, elegimos la hoja de Google Sheet de la que queremos extraer los datos y nos aparecerá la tabla en esta web.

Imagen de tabla de incidencias

La tabla que nos aparece a simple vista a cambiado muy poco, en las características podemos elegir el rango de datos de la tabla que queremos mostrar.

En parámetros avanzados podemos añadir un formato para nuestra tabla, esta apariencia estará dada por un programa que hayamos escrito en una hoja de Google Sheet, este se aplicará agregando la dirección de la celda en el que lo hayamos guardado.

Existen muchas posibilidades para crear una apariencia más atractiva para la tabla, en este caso el aspecto de los datos lo dará el siguiente texto en lenguaje html.

|  |
| --- |
| <p>-------------------------------------------------------------------------------------------</p>  </p><span style="font-size:20px">Fecha: ${"Timestamp"}</span><br>  <span style="font-size:20px">Usuario: ${"Usuario"}</span><br>  <span style="font-size:20px"><p>Incidencia: ${"Incidencia"}</span> <br>  <span style="font-size:20px">Cantidad: ${"Dinero"}</span> |

Al aplicar esta configuración veremos la tabla con el siguiente aspecto.

Imagen de tabla de incidencias como etiquetas

Una de las ventajas de utilizar Awesome table es que podemos empotrar las tablas que creemos en cualquier página web que diseñemos. Para ello hacemos clic en compartir, después en empotrar y copiamos el código que nos aparece, este lo añadiremos al de nuestra página web.

## -Crear lista de incidencias de última semana

A parte de la lista completa de incidencias también es interesante poder filtrar los datos por fecha, como por ejemplo realizar una búsqueda de las incidencias ocurridas en la última semana.

Para ello creamos otra hoja en Google Sheets y hacemos una consulta a la hoja en la que se almacenan los datos de las incidencias, la consulta será: =query(Formulario incidencias!A:D), formulario incidencias es la hoja en la que se encuentran los datos y A:D se refiere al rango de los datos que queremos extraer, que son la columnas de usuario, incidencia, dinero y timestamp.

Una vez hecha la consulta ya tenemos los datos que queremos en nuestra hoja actual, ahora solo tenemos que filtrar los datos por fecha.

Hacemos clic en el icono de crear un filtro, y veremos como nos han aparecido unos marcadores al lado de los encabezados de las columnas.



Hacemos clic en el marcador de Timestamp y le damos a filtro por condición, después a de la fecha es anterior a, y elegimos la opción de la semana pasada, le damos a ok y automáticamente aparecerán únicamente las incidencias de la semana pasada.

|  |
| --- |
| Tabla con incidencias de la semana pasada |

El objetivo de esta tabla es alojarla en la página web, al igual que en el punto anterior la apariencia de la tabla actual es bastante mejorable. Por ello realizaremos los pasos del punto anterior para transformar los datos de la hoja de Google Sheet en una awesome table, quedando de la siguiente forma.

|  |
| --- |
| Awesome table de datos de la semana pasada |

Le damos a compartir, copiamos el código de nuestra awesome table y lo pegamos junto con el código de nuestra página web.

## -Cerrar incidencias desde la web mediante el mantenimiento de las máquinas

Además de proporcionar los datos generados por el sistema también funcionara como vínculo con el personal de mantenimiento proporcionando una pantalla útil para llevar un seguimiento del mantenimiento de las maquinas.

## -Cerrar incidencias en la aplicación una vez solucionadas