









**Contenido**

[**Resumen**](#_heading=h.gjdgxs) **4**

[**Objetivo**](#_heading=h.30j0zll) **4**

[Objetivos específicos](#_heading=h.guvg40qit5iv) 4

[**Alcance**](#_heading=h.1fob9te) **4**

[**Roles y responsabilidades**](#_heading=h.3znysh7) **4**

[**Requerimientos**](#_heading=h.2et92p0) **4**

[Restricciones](#_heading=h.7byknkfu1ruj) 5

[Matriz de requerimientos](#_heading=h.qf3yk08ltiac) 5

[**Herramientas de desarrollo**](#_heading=h.pav75b3j6pcd) **8**

[**Desarrollo**](#_heading=h.sq4taunp3uzr) **8**

[Servidor local](#_heading=h.mb0muvw9feh5) 8

[Base de datos](#_heading=h.jvnhm2jrp924) 11

[Funcionalidad](#_heading=h.dbxam98qu8a4) 14

[consulta de rangos](#_heading=h.1095kcl7221j) 29

[**Referencias**](#_heading=h.kjgmml6d3bmb) **33**

# Resumen

Reporte que muestra el resultado del proyecto de desarrollo de un boletín de noticias.

# Objetivo

Desarrollar un sistema que muestre un boletín de noticias a partir de URL añadidas por el usuario.

## Objetivos específicos

* Mostrar un boletín de noticias de seguridad a partir de URL obtenidas de otros sitios.
* Descargar el boletín de noticias en formato PDF, HTML y JPG.
* Realizar un conteo de las noticias diarias, totales y en un rango específico.

# Alcance

El reporte de actividades llegará a los servidores sociales del área de Seguridad en Sistemas y al responsable inmediato de estos para notificar y validar el progreso del trabajo realizado, así como servir de apoyo para la reproducción del sistema.

# Roles y responsabilidades

A continuación, se describen los roles y responsabilidades relacionados con la actividad de realización de una máquina virtual con un servidor local que distinga direcciones seguras con https.

Responsable del área de Seguridad en Sistemas: Revisar y aprobar el desarrollo de las actividades solicitadas en la máquina virtual.

Servidor social: Ejecutar las tareas solicitadas por el responsable inmediato dentro de las máquinas virtuales.

# Requerimientos

* El sistema permitirá al usuario pegar url de sitios de noticias y agregarlos o removerlos al boletín.
* El sistema mostrará la siguiente información preliminar de cada noticia de la que se añada la URL {Imágen de la noticia | Imágen default, Título de la noticia, [Resumen de la noticia], Botón con la liga al sitio original de la noticia, [Palabras clave], URL del sitio, Nombre del sitio que publica la nota}
* El sistema permitirá al usuario descargar el listado de noticias generado en el boletín en formato HTML, PDF y JPG.
* El sistema permitirá al usuario redirigirse al sitio original de una noticia presionando un botón.
* El sistema almacenará en la base de datos la cantidad de noticias añadidas cada día por el usuario.
* El sistema mostrará la fecha actual y un conteo de las noticias añadidas a lo largo del día.
* El sistema mostrará al usuario dos calendarios que le servirán al usuario para establecer un rango de búsqueda y calcular cuantas noticias se registraron en dicho periodo.
* El sistema mostrará el total general de noticias añadidas.
* El sistema mostrará opciones de redirección a sitios de información del CSI-UNAM/CERT.

## Restricciones

* El sistema está enfocado al muestreo de noticias a partir de una URL, pero no es capaz de validar que el origen de la URL es de una noticia.
* El sistema no almacena la información de las las noticias, sólo el conteo de éstas de manera diaria, total y por periodos determinados por el usuario.
* El sistema permite el cálculo de noticias en cierto periodo, siempre y cuando la fecha de origen no sea superior a la de destino ni sobrepasen al día en transcurso.
* El sistema solamente puede descargar el boletín en formatos PDF, HTML y JPG.

## Matriz de requerimientos

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requerimiento** | **Cumplimiento** | **Observaciones** |
| **1** | La página permitirá al usuario ingresar una url de una noticia en un campo de texto vacío. | COMPLETO | El requerimiento se cumple en un 100% |
| **2** | La página permitirá mostrar el contenido preliminar de una noticia si se encuentra una url en un campo de búsqueda de noticias [Título de la noticia, Imágen, Resumen, Palabras clave, URL, Origen]. | COMPLETO | El requerimiento se cumple en un 100% |
| **3** | La página mostrará una imagen por default en caso de que la noticia original no se encuentre | COMPLETO | El requerimiento se cumple en un 100% |
| **4** | La página mostrará al usuario un apartado de información preliminar por cada noticia (url) que se ingrese. | COMPLETO | El requerimiento se cumple en un 100% |
| **5** | La página permitirá al usuario añadir tantos campos de texto vacíos para url como deseé. | COMPLETO | El requerimiento se cumple en un 100% |
| **6** | La página solo permitirá al usuario añadir url de noticias una a la vez | COMPLETO | El requerimiento se cumple en un 100% |
| **7** | La página tendrá un botón que permita añadir un nuevo campo de texto vacío | COMPLETO | El requerimiento se cumple en un 100%  El botón que permite añadir un nuevo campo de url vacío es el mismo que carga la noticia preliminar en la lista y añade un nuevo campo de url. |
| **9** | La página tendrá un botón que permita listar la vista preliminar de las noticias que se hayan ingresado una a la vez. | COMPLETO | El requerimiento se cumple en un 100%  El botón que permite añadir un nuevo campo de url vacío es el mismo que carga la noticia preliminar en la lista y añade un nuevo campo de url. |
| **10** | La página tendrá un botón que permita eliminar la lista de las url ingresadas para añadir nuevas. | COMPLETO | El requerimiento se cumple en un 100% |
| **11** | La página mostrará un botón en la vista de cada noticia preliminar que redirija a la página de donde proviene. | COMPLETO | El requerimiento se cumple en un 100% |
| **12** | La página permitirá al usuario descargar las noticias registradas en un archivo en formato html | COMPLETO | El requerimiento se cumple en un 100% |
| **13** | La página no permitirá generar un nuevo campo de url vacío si el anterior se encuentra vacío | COMPLETO | El requerimiento se cumple en un 100% |
| **14** | La página enviará el siguiente mensaje si el usuario intenta añadir un espacio de url nuevo dejando en blanco el espacio correspondiente {Por favor, llene el campo con una URL válida}. | COMPLETO | El requerimiento se cumple en un 100% |
| **15** | La página no permitirá al usuario editar ningún campo de url que ya haya sido cargado a la lista de noticias preliminares. | COMPLETO | El requerimiento se cumple en un 100% |
| **16** | La página contará con los siguientes menús {Acerca de, Servicios, Contacto} que redirigirán al usuario a las páginas de información de la CSI-UNAM/CERT | COMPLETO | El requerimiento se cumple en un 100% |
| **17** | La página contará con el siguiente menú {Home} que redirigirá a la misma página. | COMPLETO | El requerimiento se cumple en un 100% |
| **18** | La página permitirá al usuario borrar todos los elementos de la lista de url | COMPLETO | El requerimiento se cumple en un 100% |
| **19** | La página permitirá al usuario borrar cada elemento de la lista de url uno a la vez | COMPLETO | El requerimiento se cumple en un 100% |

# Herramientas de desarrollo

* Sistema Operativo
  + Debian GNU/Linux 10 (buster). Linux version 4.19.0-5-amd64 .
* Apache
  + Apache/2.4.38
* Postgresql
  + PostgreSQL 11.8.
* PHP
  + PHP 7.3.22.
* JQuery
  + JQuery 3.5.1.
  + JQuery UI 1.12.1.
* Complementos
  + jsPDF - <https://github.com/MrRio/jsPDF>
  + html2canvas - <https://html2canvas.hertzen.com/>
  + canvas2image - <https://github.com/hongru/canvas2image>
  + ForceUTF8 - <https://github.com/neitanod/forceutf8>
* Editor Atom
  + Atom 1.48.0.

1. Requisitos de almacenamiento

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Requisitos** | | Apache | Postgres (instalación básica) | Base de datos | jQuery | NotiCERT |
| **Espacio en disco** | **Mínimo** | 50 MB | 40 MB | 750 MB | 100 KB | 500 MB |
| **Recomendado** | 80 MB | 50 MB | 1.5 GB | 130 KB | 600 MB |

# Desarrollo

## Servidor local

Para comenzar a trabajar, es necesario tener un entorno de desarrollo y pruebas propicio para poner a prueba el sistema, encontrar y reparar los fallos pertinentes. Para lograrlo, se instaló la versión 2.4 de Apache (la más reciente al momento del inicio del desarrollo de este proyecto). Dentro de la terminal se ejecutaron los siguientes comandos:

$ sudo apt update && sudo apt -y upgrade

$ sudo apt -y install apache2

Una vez descargados e instalados todos los paquetes pertinentes, el servicio de apache quedará activo automáticamente, para comprobarlo, ejecutar el siguiente comando:

$ systemctl status apache2

De no estar activo, ejecutar el siguiente comando:

$ systemctl start apache2

Una vez activado el servicio de apache, es necesario tener el archivo de configuración necesario para definir las características básicas de nuestro servidor; para ello, es necesario dirigirse a la ruta:

/etc/apache2/sites-available

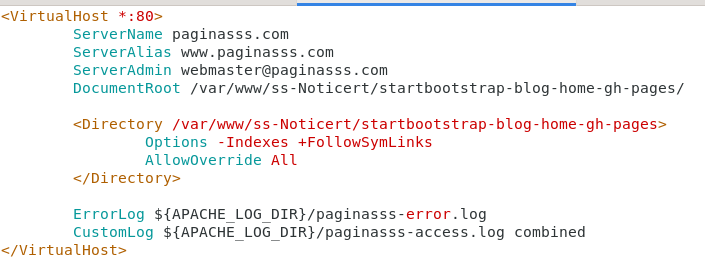
Dentro de esta ruta, encontraremos un archivo de configuración llamado “***000-default.conf***”, mismo que vamos a copiar para modificar. Es importante que como documento original no lo modifiquemos, pues es la plantilla básica de la página principal de nuestro servidor. Para copiar el archivo, ejecutar el comando:

$ sudo cp 000-default.conf pagina.com.conf

Una vez copiado el documento, vamo a modificarlo para adaptarlo a las necesidades de nuestro servidor, ejecutamos:

$ sudo vim pagina.com.conf

La terminal desplegará un espacio en el que podremos editar el contenido del archivo. Este aspecto es importante, porque determinaremos la dirección en la que nuestro sitio estará disponible, las características que tendrá y en qué archivos podremos dar seguimiento a las bitácoras de desarrollo.



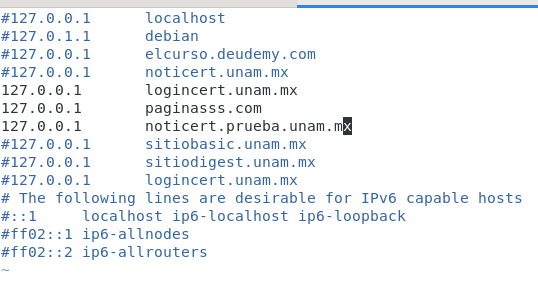
Dentro de las características del servidor, encontramos:

* *VirtualHost \*:80*: La etiqueta VirtualHost nos indica que las siguientes características serán aplicables para una página en entorno virtual local y que, específicamente, se comunicará con la red a través del puerto 80.
* *ServerName*: Que nos dará la dirección principal a la que un usuario se dirigirá cuando busque el sitio.
* *ServerAlias*: Es el nombre del sitio acompañado del prefijo “***www.***”
* *ServerAdmin*: Dirección de correo electrónico del administrador del servidor.
* *DocumentRoot*: Es la ruta de la carpeta donde se encuentran los archivos principales del sitio y lo hacen funcionar correctamente.
* La etiqueta “*Directory*” detalla un poco más a profundidad las características de los archivos que realizarán la función de index en el sitio.
* *ErrorLog*: Describe el nombre del archivo de bitácora de errores que va presentando la página durante su ejecución.
* *CustomLog*: Describe el nombre del archivo de bitácora de las funciones correctas que presenta el sitio durante su ejecución.

Ahora es necesario levantar el sitio, para ello es necesario levantar el sitio en el archivo de “hosts” con el comando:

vim /etc/hosts

En donde habilitaremos la dirección de nuestro sitio en la dirección por default del localhost:



Por último, se reinicia el servicio de apache con el comando:

$ sudo systemctl restart apache2

Si todo se ha configurado correctamente, el servicio de apache no debería arrojar ningún error y podremos acceder al sitio con la dirección que hemos registrado.

## Base de datos

La base de datos del sistema de Login es una base hecha en Postgresql; esta se encarga del almacenamiento de los usuarios y sus contraseñas. Primero se descargó e instaló Postgresql con el comando

$ sudo apt -y install postgresql

Donde “-y” nos presenta una opción de aceptar todos los paquetes que contenga la descarga. Una vez descargado e instalado, se puede verificar el estatus del servicio de postgresql con el comando:

$ systemctl status postgresql

Si no presenta ningún error, es necesario modificar la configuración por defecto del usuario y la contraseña, ya que al momento de realizar la instalación, el servicio tiene un usuario por defecto sin contraseña, por lo que ocuparlo resulta práctico, pero no tendría el nivel de seguridad mínimo que requiere nuestra base de datos, por lo que solucionamos el problema con los siguientes comandos, para crear un nuevo superusuario que nos permita tener un mejor control de la base de datos:

$ sudo -u postgres createuser --interactive

En donde “*-u*” nos indica un apartado para nombre del usuario que estamos creando, e “*--interactive*” solicita el nombre de usuario si no se especifica ninguno en la línea de comando. Posteriormente, nos pedirá completar los datos de registro del usuario:

Ingrese el nombre del rol a agregar: **postgres**

¿Será el nuevo rol un superusuario? (s/n) **S**

Con eso, se ha configurado a “postgres” como el superusuario de la base.

Para poder configurar la contraseña, primero hay que crear una base de datos para después limitar el acceso a los usuarios que quieran consumir dicha base de datos. Se ejecutan los siguientes comandos:

postgres=# \password

Ingrese la nueva contraseña:

Ingrese la contraseña nuevamente:

postgres=#

Una vez configurado el servidor de la base de datos, se configuró el archivo “***pg\_hba.conf***” que permite activar la función de validación de contraseñas con el comando:

$ sudo vim /etc/postgresql/11/main/pg\_hba.conf

Dentro del documento buscamos el apartado siguiente

# "local" is for Unix domain socket connections only

local all all md5

Entonces cambiamos el valor por defecto “md5” por el valor “peer” para activar la autenticación por contraseña, quedando de esta manera

# "local" is for Unix domain socket connections only

local all all peer

Una vez configurado el archivo, se guardan cambios y reiniciamos el servicio de postgres con el siguiente comando:

$ sudo systemctl reload postgresql

Para comprobar que todo fluye sin problemas, ejecutamos el comando:

$ systemctl status postgresql

Si el servidor está en buen estado, procedemos a crear la base de datos que utilizaremos para almacenar la información de los usuarios. Esto lo lograremos importando el documento “***pg\_ss-noticert.sql***”, donde se encuentra toda la información necesaria para la creación de la base de datos con algunos datos ejemplo dentro de ella. El archivo “.*sql*” debe guardarse en la siguiente ruta para mayor facilidad de encontrar el archivo:

/etc/postgres/

Ya que accederemos a la base de datos como administrador, no es necesario modificar ningún permiso. Ejecutamos el siguiente comando que nos permitirá acceder a la terminal de postgres, nos solicitará la contraseña que hemos creado recientemente.:

$ su - postgres

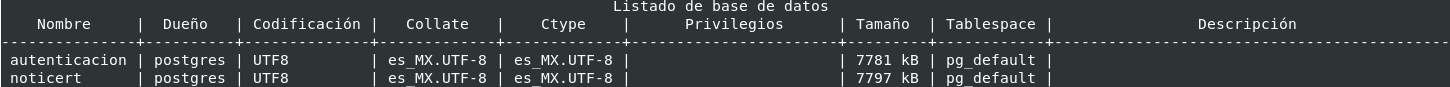
Una vez hayamos accedido, debemos importar el volcado de la base de datos con el siguiente comando:

$ psql -U postgres < pg\_ss-noticert.sql > pg\_ss-noticert.log 2>&1

Que realizará la importación de la base en la sesión del usuario “postgres” y creará un documento “.log” con el nombre que determine el usuario (en este caso también se llama “pg\_ss-noticert”) para monitorear que todo funcione con esa importación de base de datos. Para verificar que todo ha resultado correctamente, ejecutamos el siguiente comando:

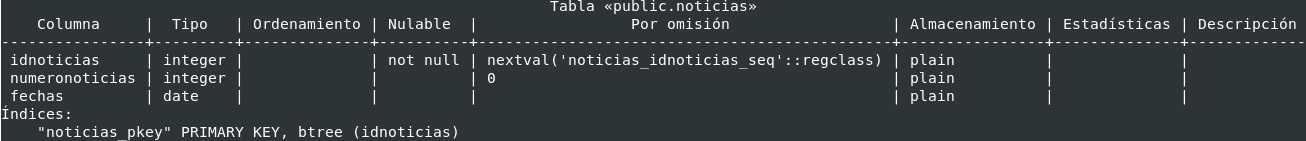
$ grep -i error pg\_ss-noticert.log

Si la importación resultó exitosa, podemos acceder sin problema. El puerto de acceso a la base es el mismo que viene por defecto al instalar el manejador: “***5432***”, dentro, encontraremos la base de datos del sistema de Noticert, misma que se encarga del almacenamiento de la cantidad de noticias registradas y de la fecha en que esto sucede.

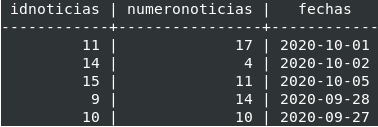


El nombre de la base es “***noticert***”, donde para poder acceder a ella, es necesario contar con el acceso a postgres es a través del usuario “***postgres***” con contraseña “***postgres***”. Dado que el desarrollo se hizo en un entorno completamente de prueba, no resultaba necesario reforzar los criterios de usuario y contraseña, pero se recomienda cambiarlos por cadenas más seguras en cuanto se suba a producción.

El cuerpo de la base de datos consta de solamente una tabla llamada “***noticias”***, compuesta por tres columnas: *idnoticias* de tipo numérico (int), *numeronoticias* de tipo numérico (int) y *fechas* de tipo fecha (date).



Dentro de la tabla se guardarán los registros de la página Noticert que se vayan agragando diariamente.



## Funcionalidad

Para la funcionalidad del sistema, se requieren de los siguientes archivos de código fuente:

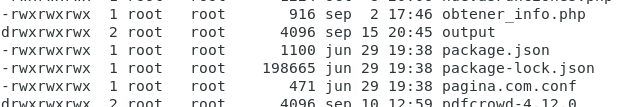
|  |  |  |
| --- | --- | --- |
| **Archivos** | **Función** | **Permisos (CHMOD)** |
| canvas2image.php | Librería que ayuda a la conversión del “div” seleccionado de html a imagen .jpg | Propietario: 4  Grupo: 4  Otros: 4 |
| cargarDatos.js | Funciones intermedias entre código principal y la consulta y consumo de la base de datos del área de calendario | Propietario: 6  Grupo: 6  Otros: 4 |
| conexionbd.php | Código que establece conexión con la base de datos. | Propietario: 6  Grupo: 4  Otros: 4 |
| consultaDiario.php | Código que ejecuta la query que consulta las noticias añadidas en el día en transcurso. | Propietario: 6  Grupo: 4  Otros: 4 |
| consultaTotal.php | Código que ejecuta la query que consulta las noticias añadidas en total | Propietario: 6  Grupo: 4  Otros: 4 |
| consultaRango.php | Código que ejecuta la query que consulta las noticias añadidas en el rango determinado por las fechas | Propietario: 6  Grupo: 4  Otros: 4 |
| crearActualizar.php | Código que ejecuta la query que crea o actualiza el registro diario de noticias conforme se van añadiendo | Propietario: 6  Grupo: 4  Otros: 4 |
| desactivarBotones.php | Función que inhabilita los botones de descarga del boletín cuando está vacío | Propietario: 4  Grupo: 4  Otros: 4 |
| DescargaNoticias.php | Aísla el fragmento del sitio que contiene la vista previa de las noticias para guardarlo en formato html | Propietario: 4  Grupo: 4  Otros: 4 |
| descargaNoticiasJPG.php | Aísla y convierte el fragmento del sitio que contiene la vista previa de las noticias para guardarlo en formato jpg | Propietario: 4  Grupo: 4  Otros: 4 |
| eliminarNoticia.php | Asociado con el botón que borra una noticia a la vez del listado. | Propietario: 4  Grupo: 4  Otros: 4 |
| Encoding.php | Código que permite forzar el protocolo UTF-8 y convierte caracteres extraños. | Propietario: 4  Grupo: 4  Otros: 4 |
| enviarUrlPrueba.php | Función que aísla la información contenida en la url para obtener los datos de noticias que nos interesan. | Propietario: 4  Grupo: 4  Otros: 4 |
| html2canvas.php | Función que habilita la conversión de un espacio de formato html y permite su conversión a imágen. | Propietario: 4  Grupo: 4  Otros: 4 |
| index.php | El código de la página principal. | Propietario: 4  Grupo: 4  Otros: 4v |
| jquery-ui.js | Librería de jQuery de uso de elementos de interfaz de usuario | Propietario: 4  Grupo: 4  Otros: 4 |
| jspdf.min.js | Librería de jQuery de uso dinámico de datos. | Propietario: 4  Grupo: 4  Otros: 4 |
| noticiashtml.php | Código que permite aislar el espacio de noticias que nos interesa imprimir en formato PDF | Propietario: 4  Grupo: 4  Otros: 4 |
| noticiasTotales.php | Consulta a la base de datos para conocer la cantidad total de noticias agregadas. | Propietario: 4  Grupo: 4  Otros: 4 |
| nuevasFunciones.php | Código que reconoce y aísla los elementos de imágen, título y metadatos de una url. | Propietario: 4  Grupo: 4  Otros: 4 |
| eliminarNoticias.php | Asociado con el botón “Eliminar listado de url”, permite borrar de la pantalla la lista actual de elementos añadidos. | Propietario: 4  Grupo: 4  Otros: 4 |
| pg\_ss-noticer.sql | Query necesaria para la creación de la base de datos. | Propietario: 4  Grupo: 4  Otros: 4 |

Se recomienda que la descarga del sistema sea ***dentro*** en la ruta “***/var/www/https/***”, ya que suele ser el proceso más común, sin embargo existe la opción de realizarla en cualquier parte, sin embargo es muy importante recalcar lo siguiente: No importa en qué ruta se almacene el proyecto siempre y cuando dicha ruta se encuentre bien definida y especificada al momento de editar el documento “***pagina.com.conf***”, pues el apartado de *DocumentRoot* y los parámetros de la etiqueta *Document* deben coincidir con la ruta en donde se ha guardado.

Otro aspecto importante es verificar, es que cada archivo cuente con los permisos de acceso pertinentes como se muestra en la tabla. Para verificar esto, nos posicionamos sobre la carpeta general de nuestro proyecto y ejecutamos el comando

$ ls -la

Que nos listará todos los archivos contenidos en la carpeta, nos mostrará qué permisos tienen y a qué usuario y grupo pertenecen. Por ejemplo:



Si alguno de los archivos no tuviera los permisos mencionados en la tabla, es pertinente cambiarlos con el comando con la siguiente sintaxis:

$ sudo CHMOD [Permiso de propietario] [Permiso de grupo] [Permiso de otros] [nombre del archivo.php | .js | etc]

Por ejemplo:

$ sudo CHMOD 664 cargarDatos.js

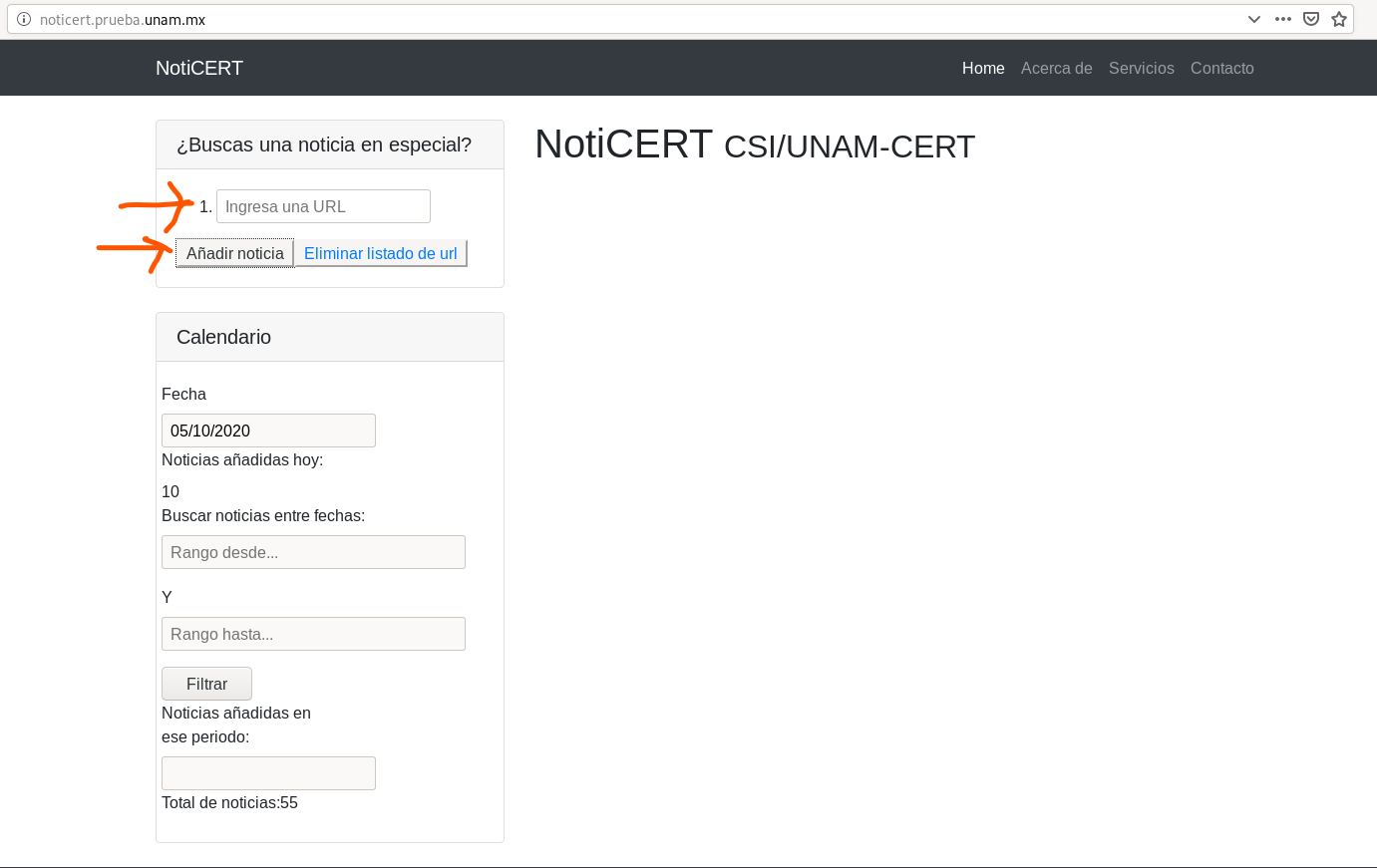
Dependerá de las necesidades del usuario el nivel de permisos que concederá. A continuación una lista de la definición de los permisos:

* 0: Sin permisos
* 1: Ejecución
* 2: Escritura
* 3: Lectura y escritura
* 4: Lectura
* 5: Lectura y ejecución
* 6: Lectura y escritura
* 7: Lectura, escritura y ejecución

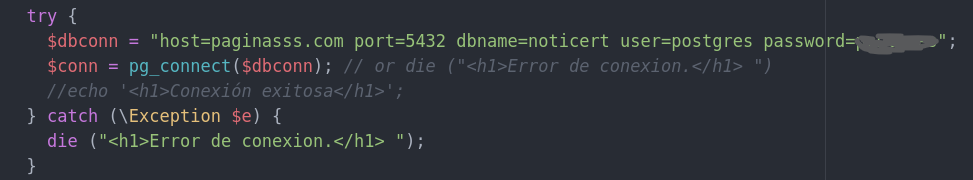
Para poder añadir una nueva noticia en la base de datos, es necesario copiar una URL de algún sitio de noticias, como se muestra a continuación:

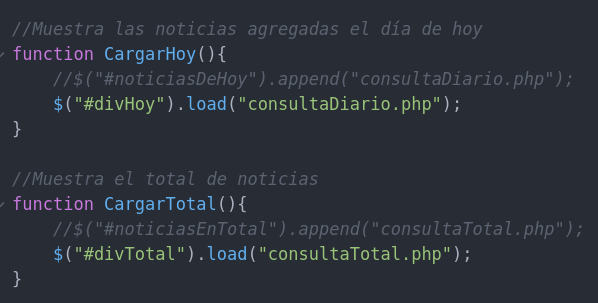


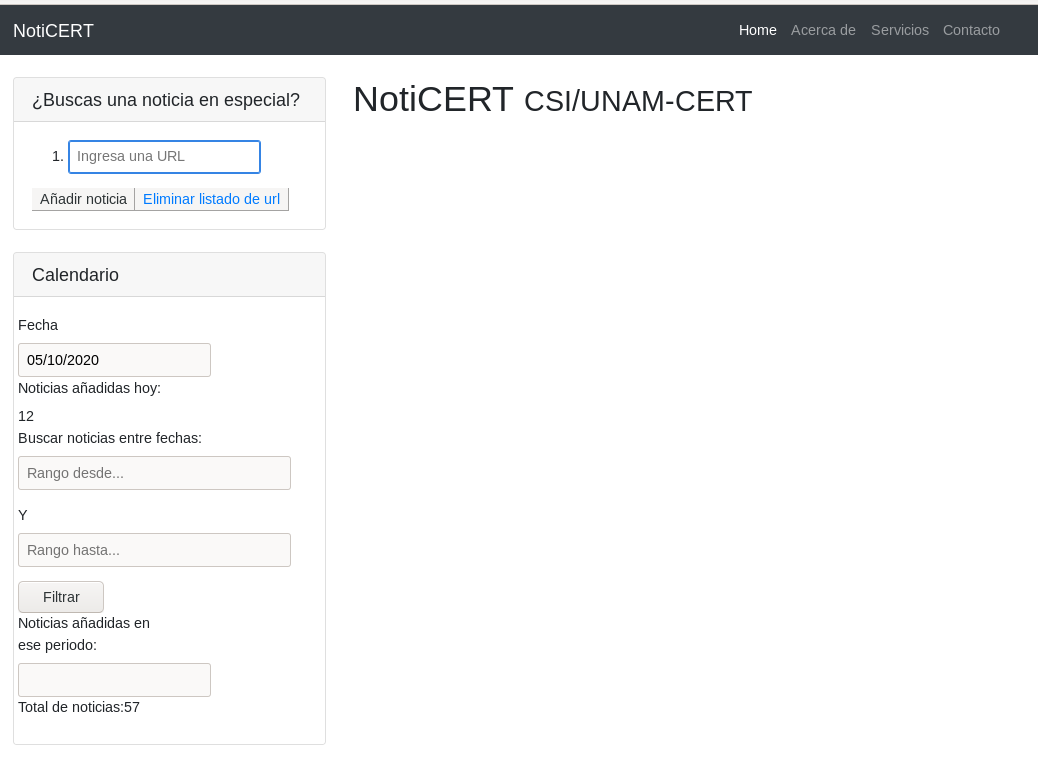
Lo pegamos en el área de búsqueda de noticias, en el espacio que dice “*Ingresa una URL*” y luego presionamos el botón de “*Añadir noticia*”. Directamente estaremos añadiendo la URL sobre un apartado input dentro de “***index.php***”. El botón para añadir noticia llama a un código de javascript que enviará la información contenida en la url al archivo “***enviarUrlPrueba.php***”, que a su vez llama al archivo “***nuevasFormulas.php***”. Ambos archivos en conjunto realizan una descomposición de los datos de la url, obteniendo la imágen de la noticia, el título de la noticia, el resumen de la noticia, palabras clave y el sitio origen de la noticia, además de que a cada elemento añadido a la lista, le asigna un identificador.,

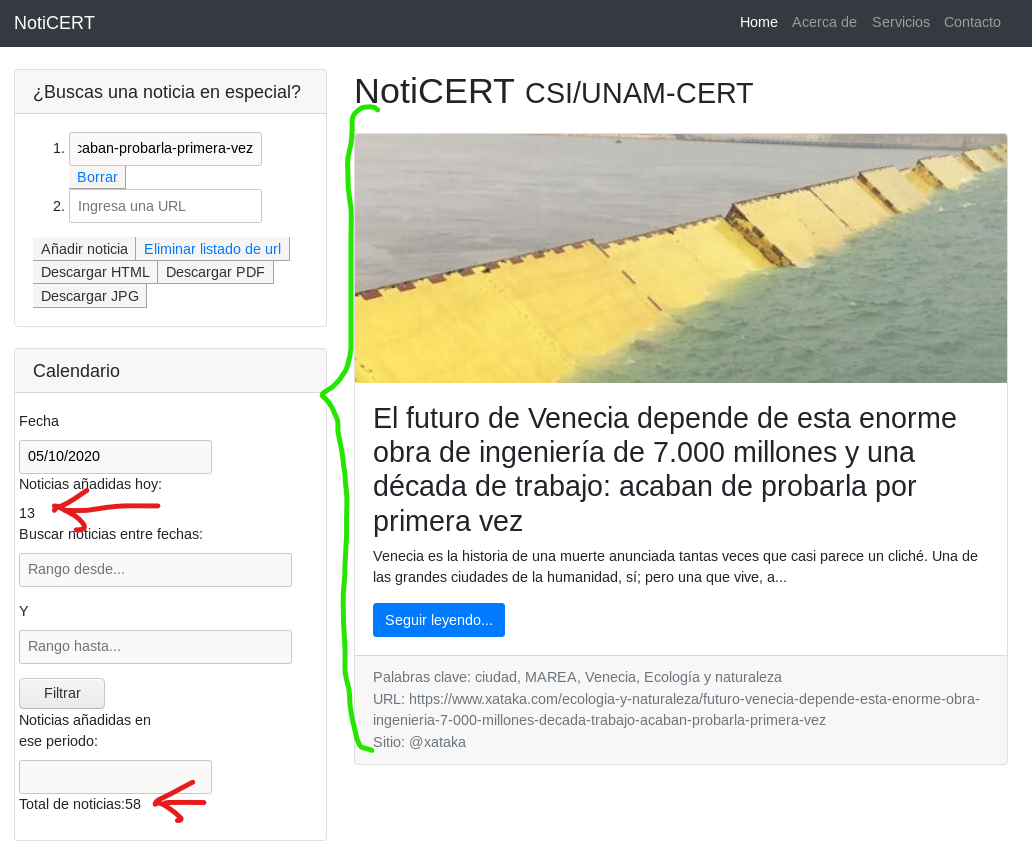


Automáticamente tras presionar el botón, la información de la noticia se mostrará en la columna de la derecha, mostrando imagen, título, resumen, liga, palabras clave, url y el sitio que la publicó; además de que el conteo de noticias diarias como el de noticias totales ha aumentado en uno. Esto último ocurre gracias a las funciones “CargarHoy” y “CargarTotal” del archivo “***cargarDatos.js***” que invoca a los archivos “***consultaDiario.php***” y “***consultaTotal.php***” respectivamente, mismos que realizan la conexión con la base de datos y realizan un cálculo de las noticias que se registraron primero durante el día actual y segundo, desde el momento en que funcionó por primera vez el sistema, devolviendo dichos resultados en los apartados de “Noticias añadidas hoy” y “Total de noticias”.

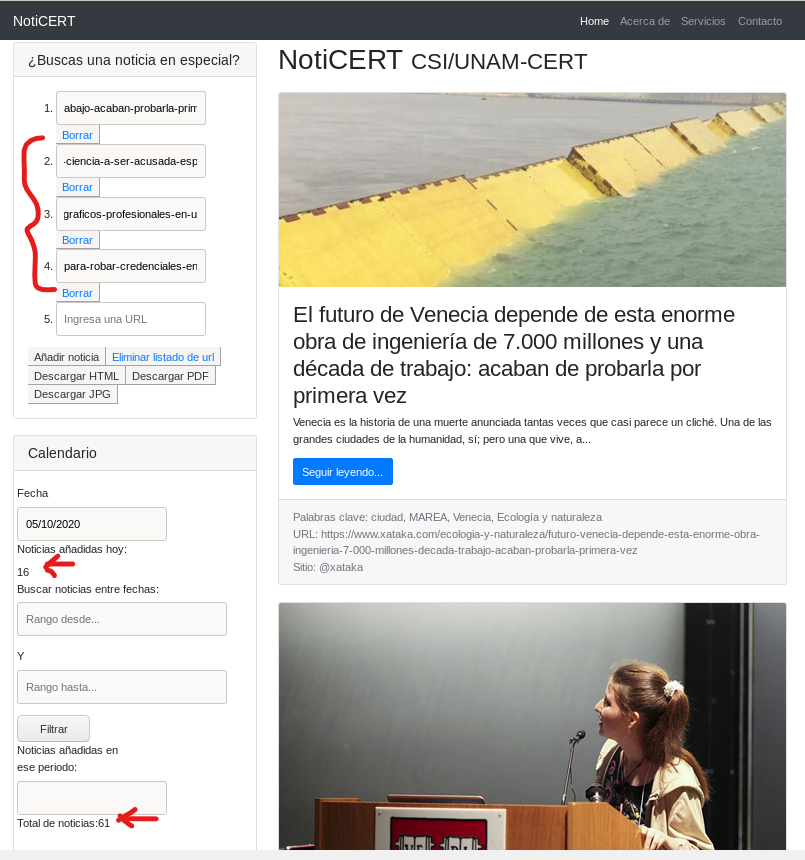




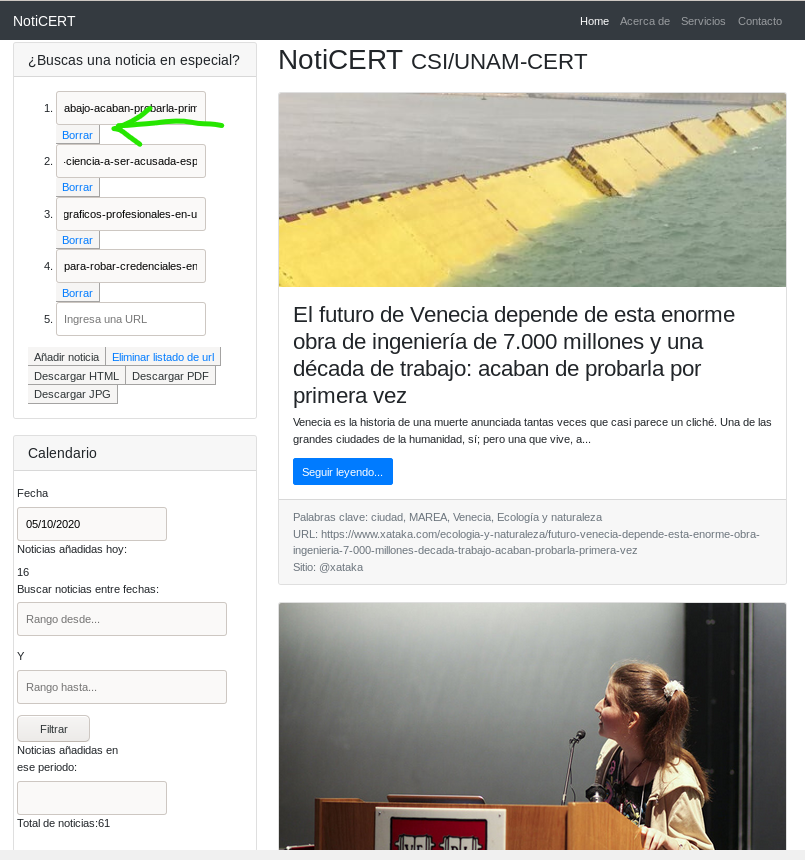




De este modo, conforme más noticias vayamos añadiendo, ambos contadores aumentarán tantas veces como añadamos una noticia; por ejemplo, habiendo añadido tres noticias más, aumentará en tres el valor de cada contador..

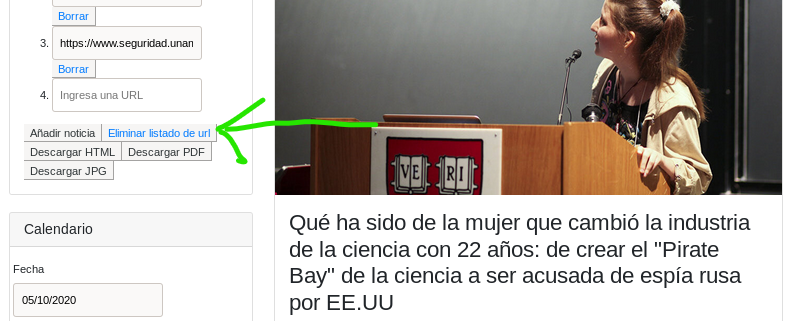


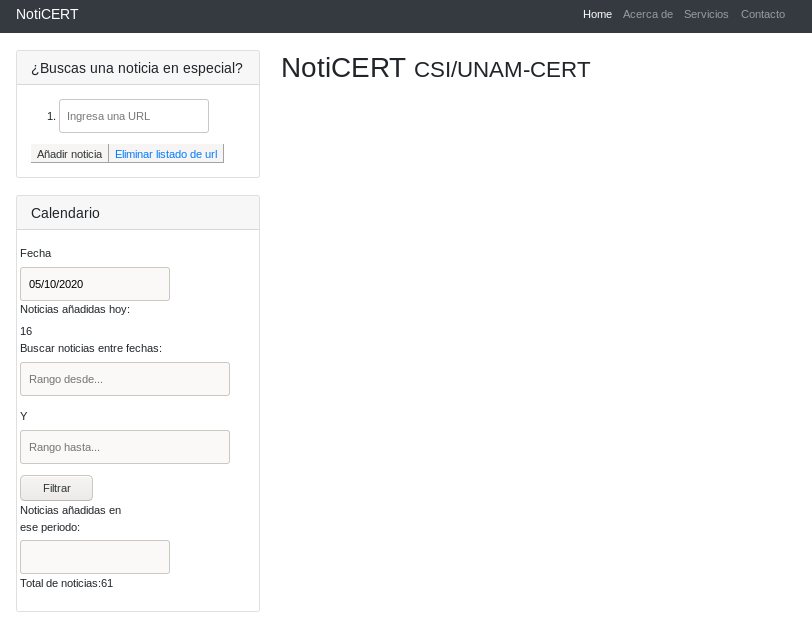
Ahora bien, si no quisiéramos eliminar alguna de las noticias ya añadidas, bastará con presionar el botón “*Borrar*” localizado al pie de cada URL ya añadida. Bastará presionarlo para eliminar la noticia que elijamos. Dicho botón invoca al archivo “***elinimarNoticia.php***” y le envía como parámetro el identificador de la noticia respectiva, de modo que el elemento (la url y, por tanto, la información de la noticia) es removido de la lista, recorriendo el listado de modo que la secuencia de los identificadores no se vea alterada.





Podremos apreciar entonces que la lista se ha reducido, una vez hemos eliminado un elemento no deseado. Por otro lado, si quisiéramos eliminar todos los elementos a la vez, basta con presionar el botón “*Eliminar listado de URL*” ; el proceso es bastante parecido, con la diferencia de que se invoca al archivo “eliminarNoticias.php” y se remueven todas las url, no una en particular.



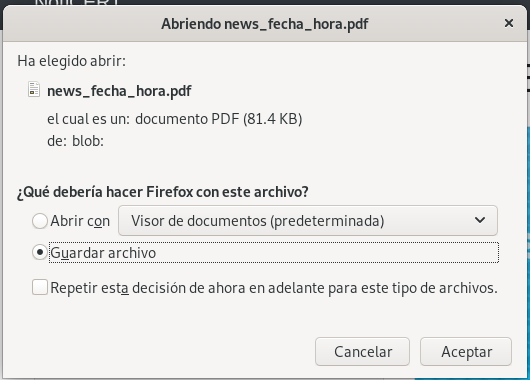
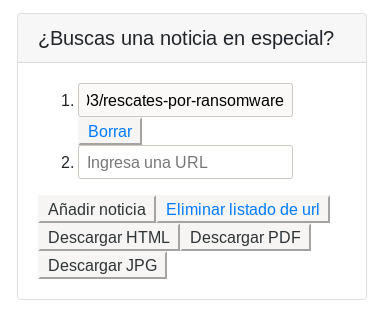


De este modo, podremos realizar un nuevo listado de noticias o cerrar la conexión del sitio hasta la siguiente ocasión que se necesite.

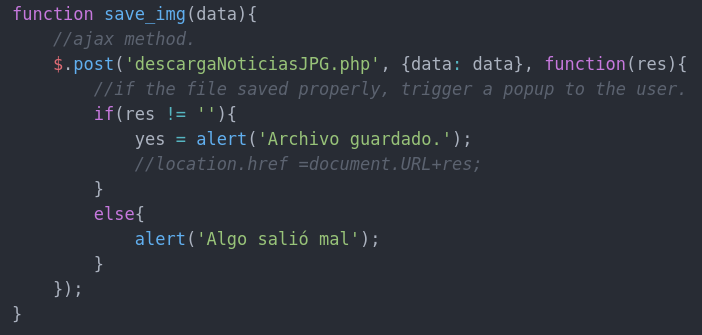
* 1. Descargas

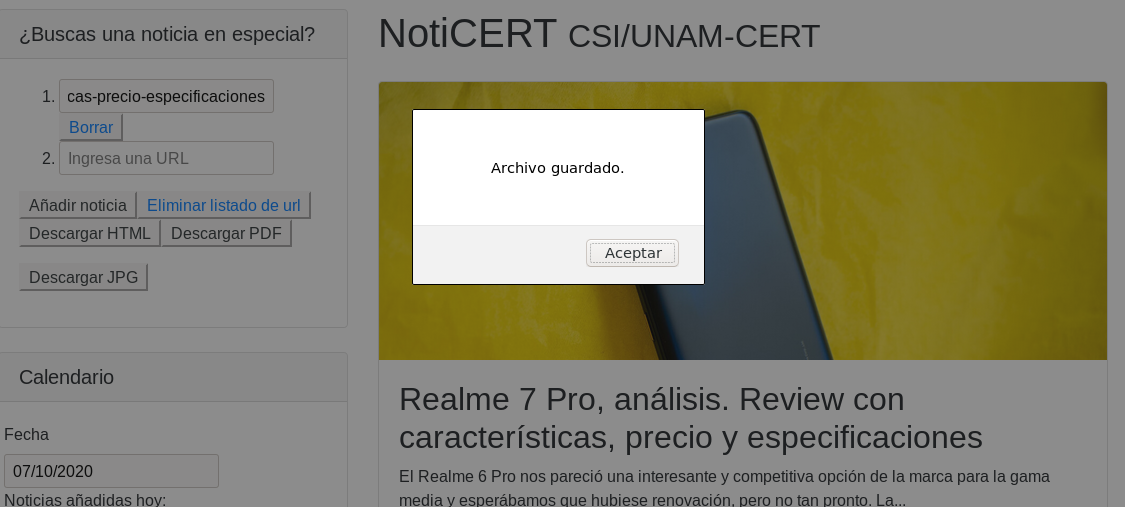
Para realizar la descarga del listado de noticias del boletín, bastará con presionar el botón correspondiente, dependiendo del formato que necesitemos almacenar: PDF, JPG o HTML.

Cada botón invoca a una función diferente, pues también necesita librerías diferentes; por ejemplo: Cuando queramos descargar nuestro boletín en formato PDF, presionaremos el botón “***DescargarPDF***” que invocará a la función “***saveDiv(‘canvas’)***”, donde “canvas” es toda el área del boletín que nos interesa guardar, en este caso, son todas las noticias que hayamos añadido, creando un nuevo PDF con ayuda de la librería ***jsPDF*** invocando a una función ajax que nos ayuda a llamar al archivo “***noticiashtml.php***” que nos ayudará a llenar las páginas necesarias del documento, dependiendo de la cantidad de noticias que tengamos añadidas en el boletín.

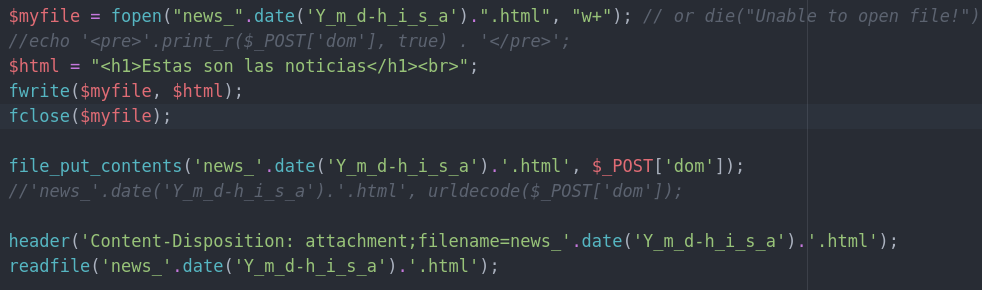


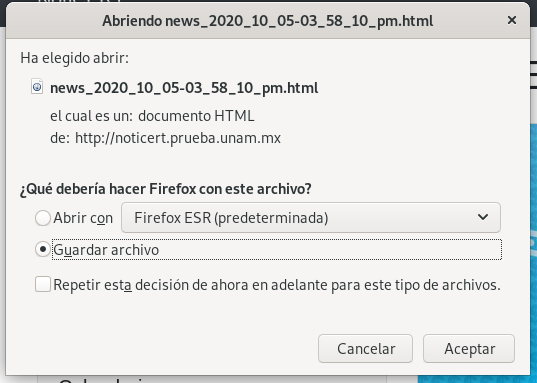
Al oprimir el botón “***DescargarJPG***” se invoca un event listener que reacciona al momento del click, éste aísla la sección de html que nos interesa convertir a imagen y lo hace gracias a la función “*html2canvas*”. Finalmente, envía los datos de la nueva imagen a la función “***save\_img(data)***” que la enviará a “***descargaNoticiasJPG.php***” para añadirle el formato de guardado en nombre y extensión, al final lo devuelve para almacenarlo en la carpeta del proyecto.





Por último, cuando se desea realizar una descarga del boletín en formato HTML, el botón “***Descargar HTML***” identifica el espacio que nos interesa guardar como archivo y lo envía como parámetro al archivo “***descargaNoticia.php***”, en donde no es necesario realizar ninguna conversión, solamente se crea el archivo html, se almacena el contenido obtenido como parámetro del botón y por último, se le da un formato de guardado y un nombre.





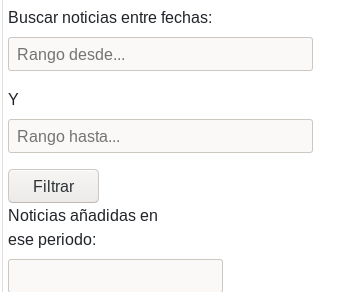
## consulta de rangos

Para poder realizar la consulta de la cantidad de noticias en cierto periodo, es necesario contar con dos campos de fechas, por lo que el documento “***cargarDatos.js***” nos será de utilidad nuevamente para utilizar la función de calendario que nos ayudará con esto.

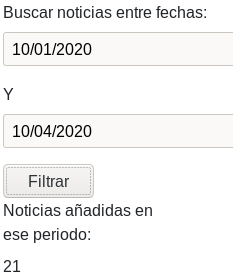


En esta función de ajax, recibimos ambos campos de fecha (*datepickerFrom* y *datepickerTo*), mismas que se enviarán al archivo “***consultaRango.php***” donde se realizarán las consultas en la base de datos para, dependiendo de las fechas y el orden en que se ingresen, mostrar un mensaje o resultado determinado, donde:

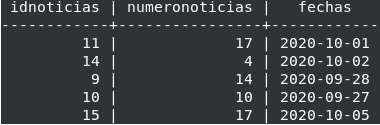
* Fecha de origen (datepickerFrom) no puede ser superior a Fecha destino (datepickerTo)
* Ni fecha de origen ni fecha de destino pueden ser superiores al día en transcurso
* Si fecha de origen es igual a fecha destino, mostrará los resultados de fecha origen
* Si fecha de origen es igual a fecha destino y a su vez es igual a la fecha actual (hoy), devolverá el resultado del registro correspondiente al día de hoy.



Así una vez hemos seleccionado nuestras fechas, bastará con dar click al botón “*Filtrar*” para que realice la consulta en “***consultaRango.php***” a la base de datos para poder traer el cálculo. Por ejemplo:

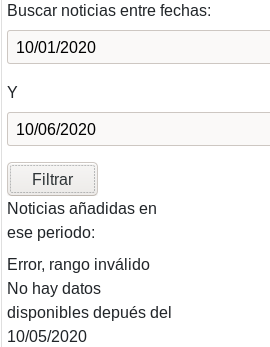


Las noticias encontradas en el periodo comprendido entre el primero de octubre y el cuatro de octubre son 21. Comparando con la base de datos:



Podemos comprobar que es cierto, solo hay dos registros comprendidos en ese rango: El del primero de octubre con 17 noticias y el del dos de octubre con 4 noticias, dando un total de 21.

En caso de que el rango no contenga nada, no mostrará nada y, en caso de que se ingrese un rango inválido, mostrará un mensaje de error como el siguiente:



Siendo la última fecha, el día en curso.

# Referencias

* Stackoverflow
  + De Paz Hernández, J. (17/sep/20). El primer PDF siempre está vacío. Lugar de publicación: es.stackoverflow.com:
  + https://es.stackoverflow.com/questions/389894/el-primer-pdf-siempre-est%c3%a1-vac%c3%ado
  + De Paz Hernández, J. (7/oct/20). Error al imprimir varias veces el mismo elemento en PDF con jsPDF. Lugar de publicación: es.stackoverflow.com:
  + https://es.stackoverflow.com/questions/395248/error-al-imprimir-varias-veces-el-mismo-elemento-en-pdf-con-jspdf
  + De Paz Hernández, J. (18/oct/20). Problemas al traer imágen de un div. Lugar de publicación: es.stackoverflow.com:
  + <https://es.stackoverflow.com/questions/390399/problemas-al-traer-im%c3%a1gen-de-un-div>
* Github
  + Hall, J. (2016). jsPDF. Lugar de publicación: github.com: https://github.com/MrRio/jsPDF
  + Hongru. (2011-2020). Canvas2image. Lugar de publicación: github.com: https://github.com/hongru/canvas2image
  + Grignoli, S. (2015-2020). Forceutf8. Lugar de publicación: github.com: <https://github.com/neitanod/forceutf8>
* Apache
  + Apache. (2020). VirtualHost Examples. Lugar de publicación: apache.org: https://httpd.apache.org/docs/2.4/vhosts/examples.html
* PHP
  + PHP. (2020). Pg\_connect. Lugar de publicación: php.net/manual: https://www.php.net/manual/es/function.pg-connect
  + PHP. (2020). GetElementById. Lugar de publicación: php.net/manual: https://www.php.net/manual/en/domdocument.getelementbyid
  + PHP. (2020). “$\_POST”. Lugar de publicación: php.net/manual: <https://www.php.net/manual/es/reserved.variables.post>
* jQuery
  + jQuery. (2020). jQueryAPI. Lugar de publicación: jquery.com: https://api.jquery.com/
  + jQuery UI. (2020). jQueryUI. Lugar de publicación: jqueryui.com: <https://jqueryui.com/datepicker/>
  + DesarrolloWeb.com. (2013). jQuery UI Datepicker. Manual de uso simple. Lugar de publicación:DesarrolloWeb.com : https://desarrolloweb.com/articulos/jquery-ui-datepicker.html
* Html2canvas
  + von Hertzen, N. (-). Html2canvas. Lugar de publicación: “https://html2canvas.hertzen.com”: https://html2canvas.hertzen.com/