

PROYECTO DE GESTIÓN DE AGUAS



AquaGest

INDICE

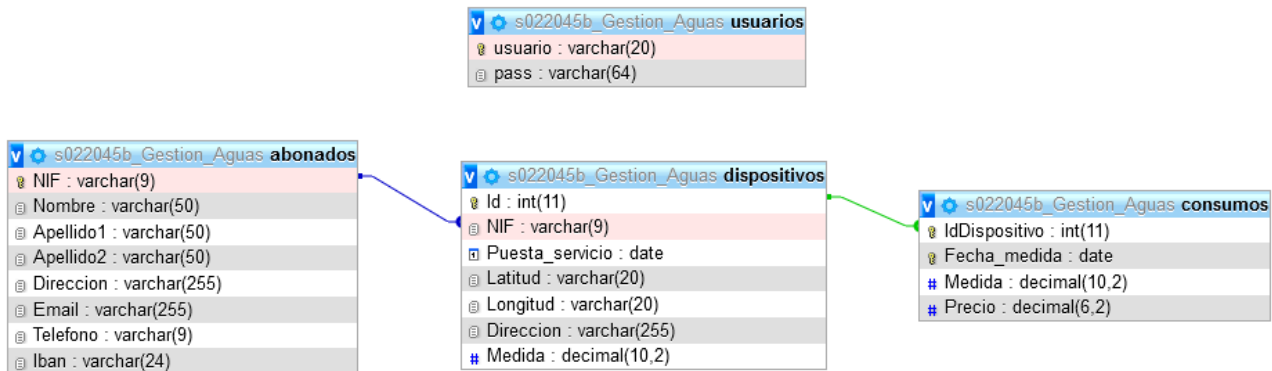
- 1. Descripción del proyecto	3
- 2. Descripción de la base de datos	3
- 3. Descripción de la interfaz	6
- 4. Creación de un nuevo abonado	9
- 5. Primer o Último registro abonados	15
- 6. Siguiente o Anterior abonados	15
- 7. Listado abonados	16
- 8. Modificar abonados	17
- 9. Borrar abonados	17
- 10. Creación de un nuevo dispositivo	18
- 11. Listado dispositivos	20
- 12. Mapa dispositivos	21
- 13. Generador de consumos	21
- 14. Generador de fechas	24
- 15. Facturación	28
- 16. Emisión correo electrónico	35
- 17. Despliegue de la aplicación	38

1. Descripción del proyecto:

La idea del proyecto es gestionar el consumo de agua de los habitantes de un municipio mediante caudalímetros inteligentes que viertan sus datos en una base de datos por tecnología inalámbrica. En una primera implementación solo estarán disponibles las opciones para el administrador del sistema (usuario = "admin", contraseña = "admin"), porque desconozco el tiempo que me va a llevar la implementación y debe ser un proyecto viable que posteriormente desarrollaré ampliando sus funcionalidades.

2. Descripción de la base de datos:

La base de datos creada presenta la siguiente estructura.



El script que genera la base de datos es el siguiente:

```
drop database if exists s022045b_Gestion_Aguas;
create database if not exists s022045b_Gestion_Aguas;

use s022045b_Gestion_Aguas;

-- Tabla usuarios.
create table usuarios(
    usuario varchar(20) primary key,
    pass varchar(64) not null
)engine=innnoDB default charset=latin1 collate=latin1_spanish_ci;

-- Creación del usuario admin.
insert into usuarios select 'admin' , sha2('admin',256);
```

```

-- Tabla abonados.
create table if not exists abonados(
    NIF varchar(9) primary key,
    Nombre varchar(50) not null,
    Apellido1 varchar(50) not null,
    Apellido2 varchar(50) not null,
    Direccion varchar (255) not null,
    Email varchar(255) not null,
    Telefono varchar(9) not null,
    Iban varchar(24) not null
) engine=innnoDB default charset=latin1 collate=latin1_spanish_ci;

-- Tabla dispositivos.
create table if not exists dispositivos(
    Id int auto_increment,
    NIF varchar(9),
    Puesta_servicio date not null,
    Latitud varchar(20) not null,
    Longitud varchar(20) not null,
    Direccion varchar (255) not null,
    Medida decimal(10,2) not null,
    primary key (Id),
    constraint fk_nif foreign key (NIF) references abonados(NIF) on delete cascade on update
cascade
) engine=innnoDB default charset=latin1 collate=latin1_spanish_ci;

-- Tabla consumos.
create table if not exists consumos(
    IdDispositivo int,
    Fecha_medida date,
    Medida decimal(10,2) not null,
    Precio decimal(6,2) not null,
    primary key (IdDispositivo, Fecha_Medida),
    constraint fk_id_dispositivo foreign key (IdDispositivo) references dispositivos(Id) on delete
cascade on update cascade
) engine=innnoDB default charset=latin1 collate=latin1_spanish_ci;

```

La tabla usuarios contiene el usuario (clave primaria) y la contraseña cifrada a verificar en el acceso a la aplicación.

La tabla abonados cuya clave primaria es el NIF, recoge los siguientes datos de los abonados: NIF, nombre, primer apellido, segundo apellido, dirección, e-mail, teléfono e IBAN. La segunda tabla dispositivos cuya clave primaria Id es un número auto-incremental contiene los datos siguientes del dispositivo: Id, NIF del abonado con el que se relaciona el dispositivo, fecha de la puesta en servicio, latitud y longitud de la ubicación, dirección y la última medida que el dispositivo ha volcado a la base de datos por tecnología inalámbrica (no dispongo de estos dispositivos por lo que se ha implementado en el código una tarea que va actualizando periódicamente el consumo generado aleatoriamente). La tercera tabla consumos cuya clave primaria es el Id del dispositivo y la fecha de la toma del consumo, contiene además la medida del dispositivo en ese momento y el precio que puede variar en función del tiempo, para lo que se ha implementado una constante para tal efecto en el código.

```
const PRECIO_METRO_CUBICO = 3.5 //Constante que define el importe facturado por cada metro cubico de agua.
```

Por razones de consistencia de la información y evitar disfunciones por una manipulación incorrecta de los datos, en principio solo se ha dispuesto la manipulación de los abonados y dispositivos.

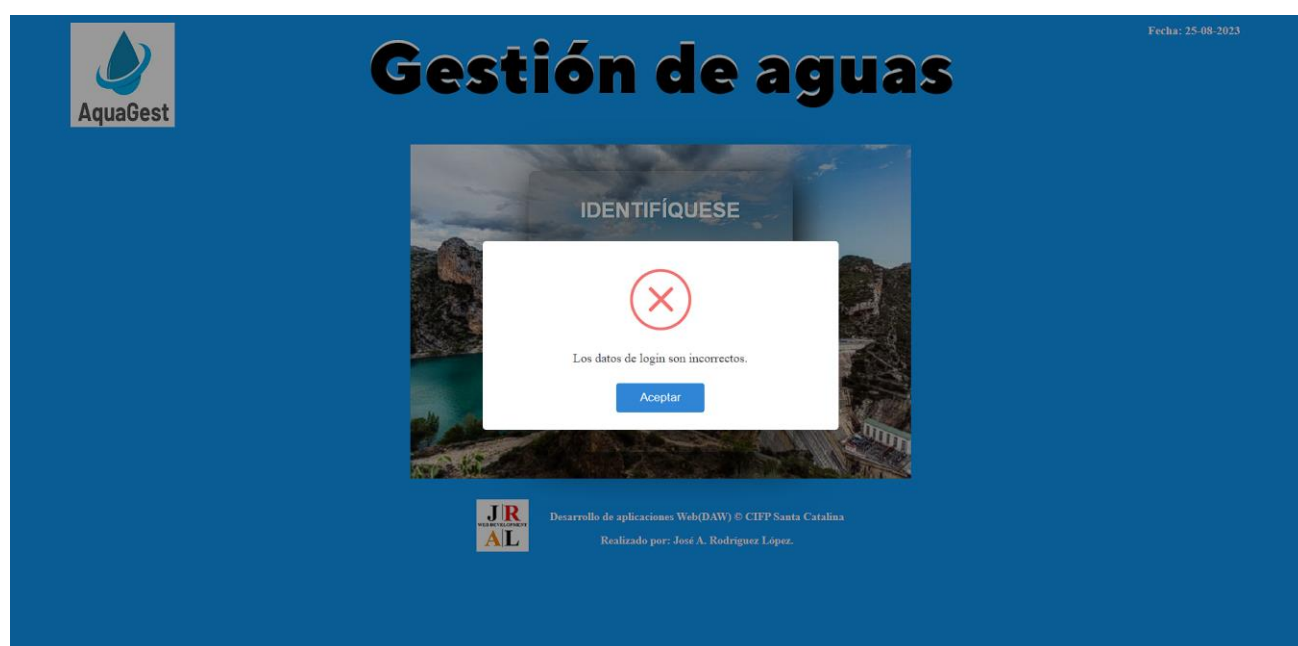
3. Descripción de la interfaz:

Página index.php::



Presenta en el encabezado a la izquierda el logotipo del servicio de aguas municipales. A la derecha arriba, muestra la fecha que va variando periódicamente con el objetivo de poder simular el funcionamiento del sistema sin tener que esperar a las fechas reales de facturación (más adelante explicaré cómo funciona). En el centro se encuentra el formulario de “login”. El usuario habilitado es “admin” y la contraseña “admin”.

Si el acceso no es correcto muestra una ventana emergente indicando el error producido.



Tras introducir el usuario y contraseña correcta se accede a la página principal.

Página principal.html:

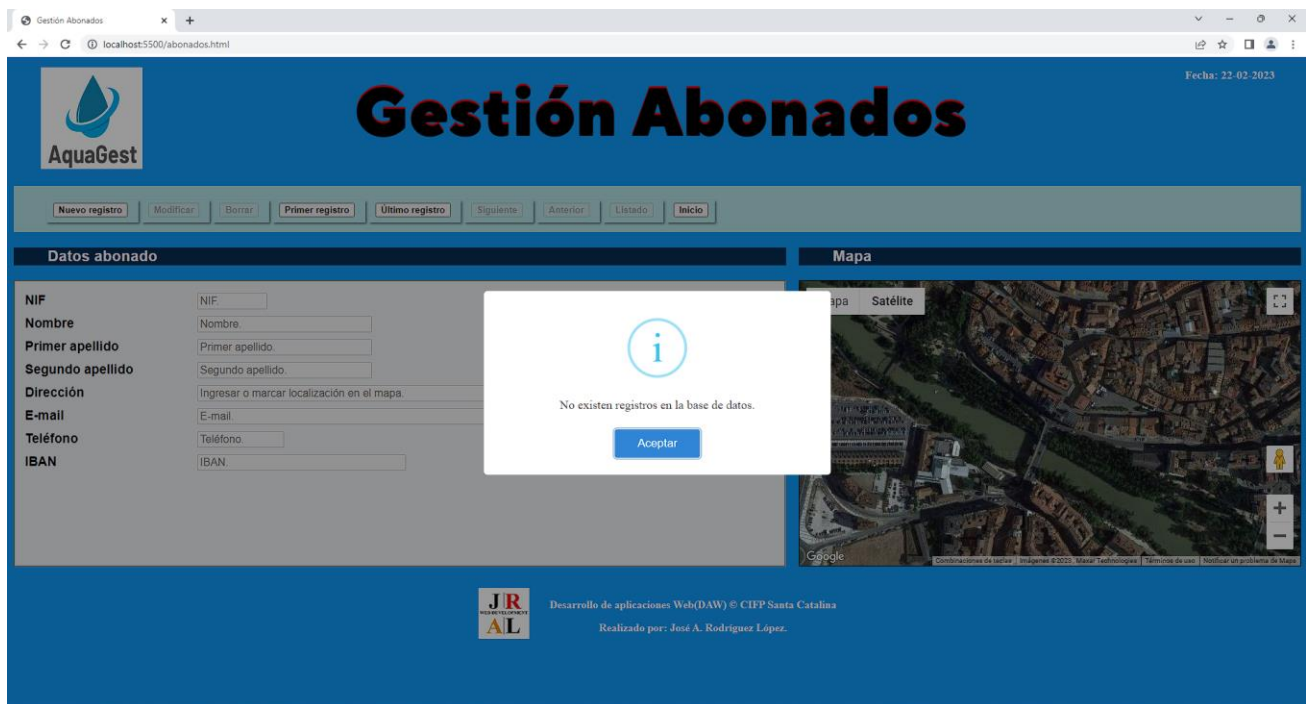


En la página principal además del encabezado común para todas las páginas, dispone en la parte central de dos botones que dan paso a la gestión de abonados o dispositivos según se estime.

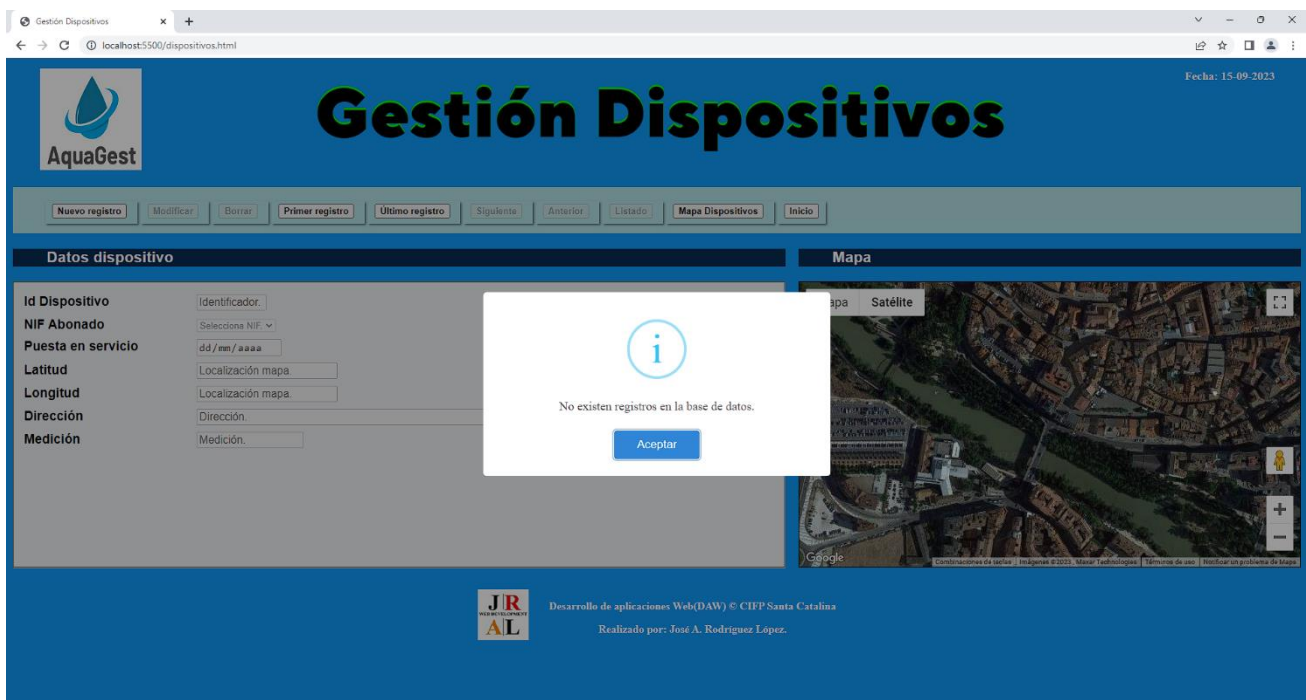
Página abonados.html:

En la página de abonados además del encabezado común para todas las páginas, dispone de un menú con las opciones típicas de manipulación de los datos (CRUD), una opción que descarga un documento PDF con los datos relativos a los abonados y otra para volver a la página de inicio.

Por otra parte, presenta a la izquierda el formulario con los datos a rellenar o consultar de un abonado. A la derecha presenta un mapa en el que por comodidad podemos introducir la dirección haciendo click en la dirección del abonado.



Página dispositivos.html:

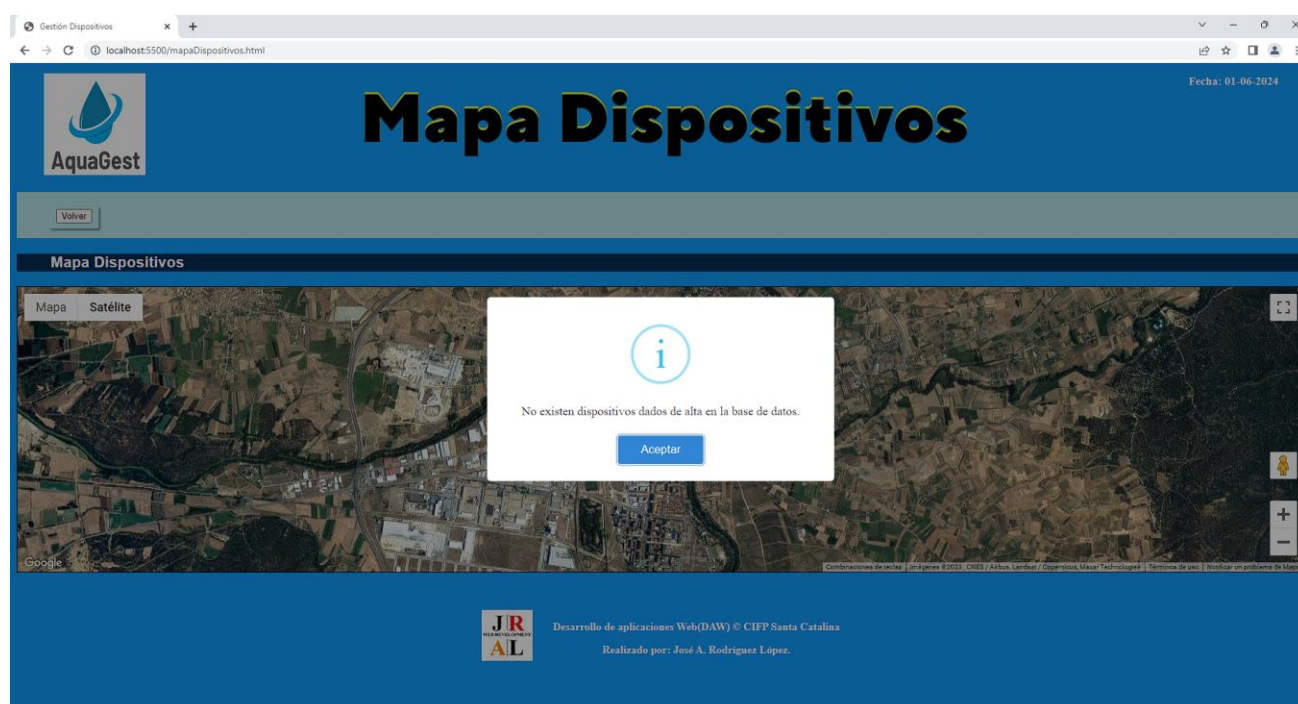


La interfaz de dispositivos presenta una apariencia similar a la interfaz de abonados con la finalidad que una vez se ha aprendido el uso de una de ellas es similar en toda la aplicación, lo que favorece la

usabilidad. Además de las opciones anteriormente explicadas el menú dispone de la opción Mapa Dispositivos que dará paso a la siguiente página.

Página mapaDispositivos.html:

Esta interfaz muestra mediante marcadores en un mapa todos los dispositivos dados de alta en el sistema. Estos marcadores al hacer click en cada uno de ellos mostrará la información relativa al dispositivo. El menú presenta la opción de vuelta a la página de gestión de dispositivos.



4. Creación de un nuevo abonado:

Pulsando sobre la opción de nuevo registro, se habilitará la grabación de este, pudiendo introducir la información solicitada.



En el NIF valida que existan 8 cifras y la letra que debe coincidir con una letra válida, ya que se ha implementado este tipo de verificación.

```
//Valida los datos del abonado.
function validarAbonado(evt) {
    let mensaje = ''
    let validado = true

    //Valida el NIF.
    if (
        (evt.target.id === 'iNIF' ||
         evt.target.id === 'bGrabar' ||
         evt.target.id === 'bModificar') &&
        validado
    ) {
        let patron = /^[0-9]{8}[A-Z]{1}$/
        let NIF = iNIF.value.trim().toUpperCase()
        let resultado = patron.test(NIF)
        if (!resultado || !validarNIF(NIF)) {
            mensaje = 'NIF no válido. 8 cifras y 1 letra. Valida la letra del NIF.'
            validado = false
        }
    }
}
```

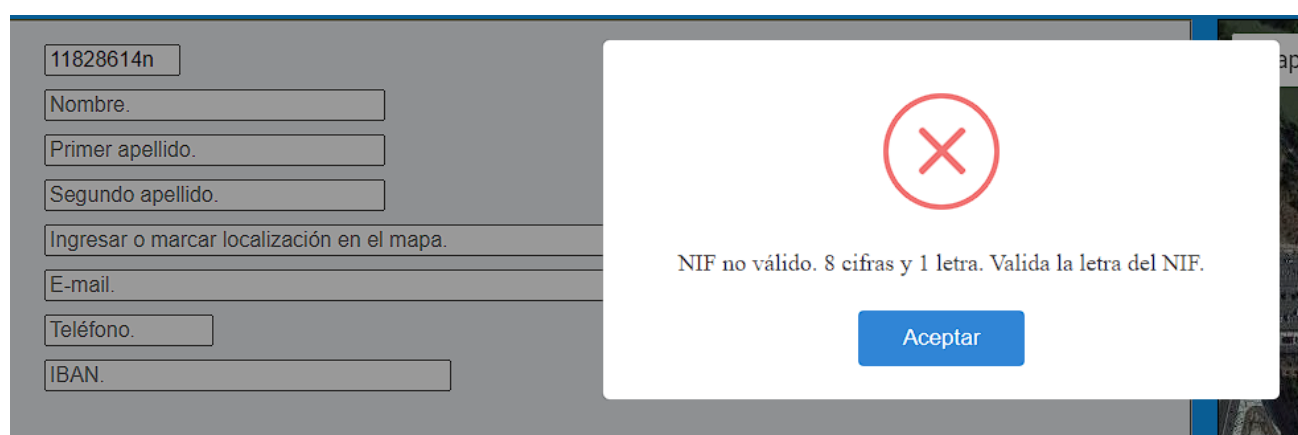
```
//Validar letra del NIF.
function validarNIF(NIF) {
    let letrasNIF = [
        'T',
```

```

    'R',
    'W',
    'A',
    'G',
    'M',
    'Y',
    'F',
    'P',
    'D',
    'X',
    'B',
    'N',
    'J',
    'Z',
    'S',
    'Q',
    'V',
    'H',
    'L',
    'C',
    'K',
    'E',
  ]
  let letra = NIF.toUpperCase().substring(8)
  let numero = parseInt(NIF.substring(0, 8))
  return letrasNIF[numero % 23] === letra
}

```

Si no se cumple alguna de las validaciones el sistema lo notificará mediante una ventana emergente.



The image shows a web form with several input fields: a text field containing '11828614n', and fields for 'Nombre.', 'Primer apellido.', 'Segundo apellido.', 'Ingresar o marcar localización en el mapa.', 'E-mail.', 'Teléfono.', and 'IBAN.'. A modal dialog box is overlaid on the right side of the form. The dialog has a white background and a red 'X' icon in a circle at the top. The text inside the dialog reads: 'NIF no válido. 8 cifras y 1 letra. Valida la letra del NIF.' At the bottom of the dialog is a blue button labeled 'Aceptar'.

El nombre valida que puede ser un nombre compuesto y solo admite los caracteres normales del idioma castellano.

```

//Valida el Nombre.
if (

```

```

    (evt.target.id === 'iNombre' ||
      evt.target.id === 'bGrabar' ||
      evt.target.id === 'bModificar') &&
    validado
  ) {
    let patron = /^[a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]+(\s([a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]+))?$ /
    let resultado = patron.test(iNombre.value.trim())
    if (!resultado) {
      mensaje =
        'Nombre no válido. Caracteres admitidos: "a-zA-ZñÑáéíóúÁÉÍÓÚüÜ". Nombres
compuestos por un máximo de 2 subcadenas separadas por un espacio.'
      validado = false
    }
  }
}

```

En los apellidos se valida que se utilicen los caracteres validos mas comunes en el idioma castellano.

```

//Valida el Apellido1.
if (
  (evt.target.id === 'iApellido1' ||
    evt.target.id === 'bGrabar' ||
    evt.target.id === 'bModificar') &&
  validado
) {
  let patron = /^[a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]+$ / //Caracteres válidos en apellido1.
  let resultado = patron.test(iApellido1.value.trim())
  if (!resultado) {
    mensaje =
      'Primer apellido no válido. Caracteres admitidos: "a-zA-ZñÑáéíóúÁÉÍÓÚüÜ".'
    validado = false
  }
}

//Valida el Apellido2.
if (
  (evt.target.id === 'iApellido2' ||
    evt.target.id === 'bGrabar' ||
    evt.target.id === 'bModificar') &&
  validado
) {
  let patron = /^[a-zA-ZñÑáéíóúÁÉÍÓÚüÜ]+$ / //Caracteres válidos en apellido2.
  let resultado = patron.test(iApellido2.value.trim())
  if (!resultado) {
    mensaje =
      'Segundo apellido no válido. Caracteres admitidos: "a-zA-ZñÑáéíóúÁÉÍÓÚüÜ".'
    validado = false
  }
}
}

```

La dirección válida que no esté vacía. Para su introducción nos podemos asistir del mapa pulsando en su localización o introducirla a mano a voluntad del usuario.

```
//Valida la Dirección.
if (
  (evt.target.id === 'iDireccion' ||
    evt.target.id === 'bGrabar' ||
    evt.target.id === 'bModificar') &&
  validado
) {
  let resultado = iDireccion.value.trim()
  if (resultado === '') { //La dirección no puede estar vacía.
    mensaje = 'La dirección no puede estar vacía.'
    validado = false
  }
}
```

En E-mail valida que cumpla la expresión regular indicada a continuación en el código. Para pruebas de depuración se recomienda introducir una cuenta válida porque la notificación de facturación será enviada a dicha cuenta.

```
//Valida el e-mail.
if (
  (evt.target.id === 'iMail' ||
    evt.target.id === 'bGrabar' ||
    evt.target.id === 'bModificar') &&
  validado
) {
  let patron = /^[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/+=?^_`{|}~-
  ]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?$/
  let resultado = patron.test(iMail.value.trim())
  if (!resultado) {
    mensaje = 'La dirección de correo electrónico no es válida.'
    validado = false
  }
}
```

El teléfono valida que es un número español, es decir, puede disponer del prefijo de país y debe empezar por 6,7,8 y 9 seguido de otras 8 cifras.

```
//Valida el teléfono.
if (
  (evt.target.id === 'iTelefono' ||
    evt.target.id === 'bGrabar' ||
    evt.target.id === 'bModificar') &&
```

```

    validado
  ) {
    let patron = /^(\\+34|0034|34)?[6789]\\d{8}$/ //Pueden llevar el prefijo de España
    y empiezan por 6, 7, 8 o 9 mas 8 cifras.
    let resultado = patron.test(iTelefono.value.trim())
    if (!resultado) {
      mensaje = 'El teléfono introducido no es válido. Pueden llevar el prefijo de
    España y empiezan por 6, 7, 8 o 9 mas 8 cifras.'
      validado = false
    }
  }
}


```

El IBAN valida que empiece por dos letras seguido de 22 cifras.

```

//Valida el Iban.
if (
  (evt.target.id === 'iIban' ||
    evt.target.id === 'bGrabar' ||
    evt.target.id === 'bModificar') &&
  validado
) {
  let patron = /^[a-zA-Z]{2}(\\d{22})$/ //2 letras y 22 dígitos.
  let resultado = patron.test(iIban.value.trim().toUpperCase())
  if (!resultado) {
    mensaje = 'El IBAN no es válido. 2 letras del código de país y 22 cifras.'
    validado = false
  }
}
}

```



Gestión Abonados

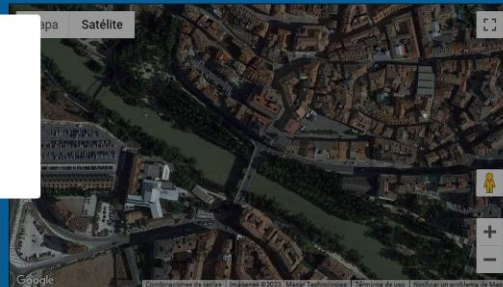
Fecha: 19-01-2032


Nuevo registro
Modificar
Borrar
Primer registro
Último registro
Siguiente
Anterior
Listado
Inicio

Datos abonado

NIF	11828614j
Nombre	jose antonio
Primer apellido	rodriguez
Segundo apellido	lópez
Dirección	Pl. Trigo, 8, 09400 Aranda de Duero, Burgos, España
E-mail	s022045b_rodriguez@informaticasc.com
Teléfono	689688888
IBAN	es11111111111111111111


Mapa





Registro grabado correctamente.

Aceptar




Desarrollo de aplicaciones Web(DAW) © CIEP Santa Catalina

Realizado por: José A. Rodríguez López.

5. Primer o Último registro abonados:

Haciendo click sobre Primer registro o Último registro, presenta en la interfaz el primer o último registro respectivamente ordenados por el NIF.



Gestión Abonados

Nuevo registro

Modificar

Borrar

Primer registro

Último registro

Siguiente

Anterior

Listado

Inicio

Datos abonado

NIF

11828614J

Nombre

Fernando

Primer apellido

Alonso

Segundo apellido

Carrera

Dirección

Pl. Trigo, 8, 09400 Aranda de Duero, Burgos, España

E-mail

s022045b_rodriguez@informaticasc.com


Teléfono


689111111

IBAN

ES1111111111111111111111

Mapa





Desarrollo de aplicaciones Web(DAW) © CIFF Santa Catalina

Realizado por: José A. Rodríguez López.

6. Siguiente o Anterior abonados:

Muestra el siguiente o anterior NIF si existe, en caso contrario lanza una notificación.


AquaGest

Gestión Abonados

Fecha: 11-03-2033

Nuevo registro

Modificar

Borrar

Primer registro

Último registro

Siguiente

Anterior

Listado

Inicio

Datos abonado

NIF

11828615Z

Nombre

José Antonio

Primer apellido

Rodríguez

Segundo apellido

López

Dirección

C. Burgo de Osma, 3D, 09400 Aranda de Duero, Burgos, España

E-mail

s022045b_rodriguez1@informaticasc.com

Teléfono

999999999

IBAN


ES22222222222222222222222222222222

Mapa




JR
RESOLUCIONES
AL

Desarrollo de aplicaciones Web(DAW) © CIPF Santa Catalina
Realizado por: José A. Rodríguez López.



Gestión Abonados

Fecha: 30-04-2033

Nuevo registro

Modificar

Borrar

Primer registro

Último registro

Siguiente

Anterior

Listado

Inicio

Datos abonado

NIF

11828617Q

Nombre

Pedro

Primer apellido

Sanchez

Segundo apellido

Arnaiz

Dirección

C. Hospicio, 2, 09400 Aranda de Duero, Burgos, España

E-mail

j062667@gmail.com

Teléfono

633333333

IBAN

ES06666666666666666666

Mapa

Mapa

Satélite

i

No existe un registro posterior.

Aceptar

JR


AL

Desarrollo de aplicaciones Web(DAW) © CIPF Santa Catalina

Realizado por: José A. Rodríguez López.

7. Listado abonados:

Esta opción descarga un documento PDF con todos los abonados registrados en el sistema.




Listado Abonados

NIF	Nombre	Apellido 1	Apellido 2	email	Dirección	Teléfono	IBAN
11828614J	Fernando	Alonso	Carrera	s022045b_rodriguez@informaticasc	Pl. Trigo, 8, 09400 Aranda de Duero, Burgos, España	689111111	ES111111111111111111111111
11828615Z	José Antonio	Rodríguez	López	s022045b_rodriguez1@informaticasc	Car. Burgo de Osma, 3D, 09400 Aranda de Duero, Burgos, España	999999999	ES222222222222222222222222
11828616S	Pedro	Duque	Astronauta	s022045b_rodriguez2@informaticasc	C. Isilla, 13, 09400 Aranda de Duero, Burgos, España	666666666	ES222222222222222222212222
11828617Q	Pedro	Sanchez	Arnaiz	j062667@gmail.com	C. Hospicio, 2, 09400 Aranda de Duero, Burgos, España	633333333	ES666666666666666666666666

GESTION DE AGUAS-JOSE A. RODRIGUEZ LOPEZ

Pg.: 1

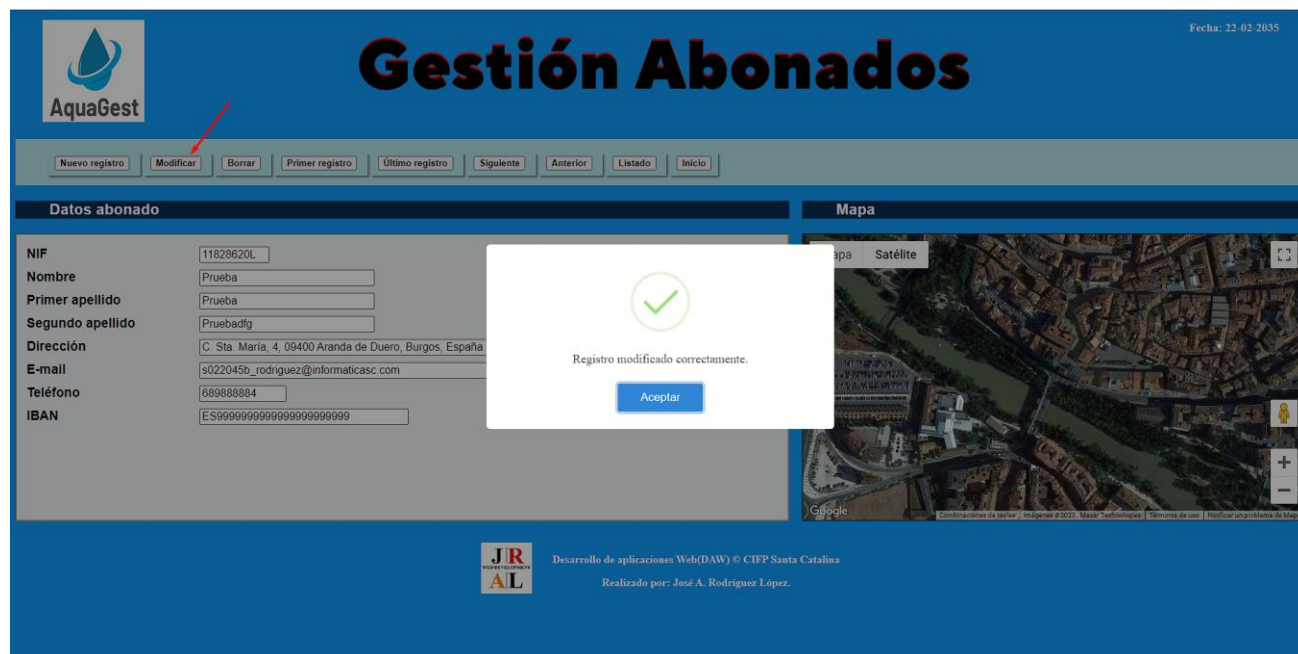


Proyecto de Gestión de Aguas Realizado por: José A. Rodríguez López

Pág. 16

8. Modificar abonados:

En esta opción se pueden modificar los campos de un abonado menos lógicamente el NIF. Deberá de cumplir los mismos criterios de validación que un nuevo registro.



Fecha: 22-02-2035

Gestión Abonados

Nuevo registro **Modificar** Borrar Primer registro Último registro Siguiente Anterior Listado Inicio

Datos abonado

NIF: 11828620L
Nombre: Prueba
Primer apellido: Prueba
Segundo apellido: Pruebadfg
Dirección: C. Sta. María, 4, 09400 Aranda de Duero, Burgos, España
E-mail: s022045b_rodriguez@informaticasc.com
Teléfono: 689888884
IBAN: ES9999999999999999999999

Mapa

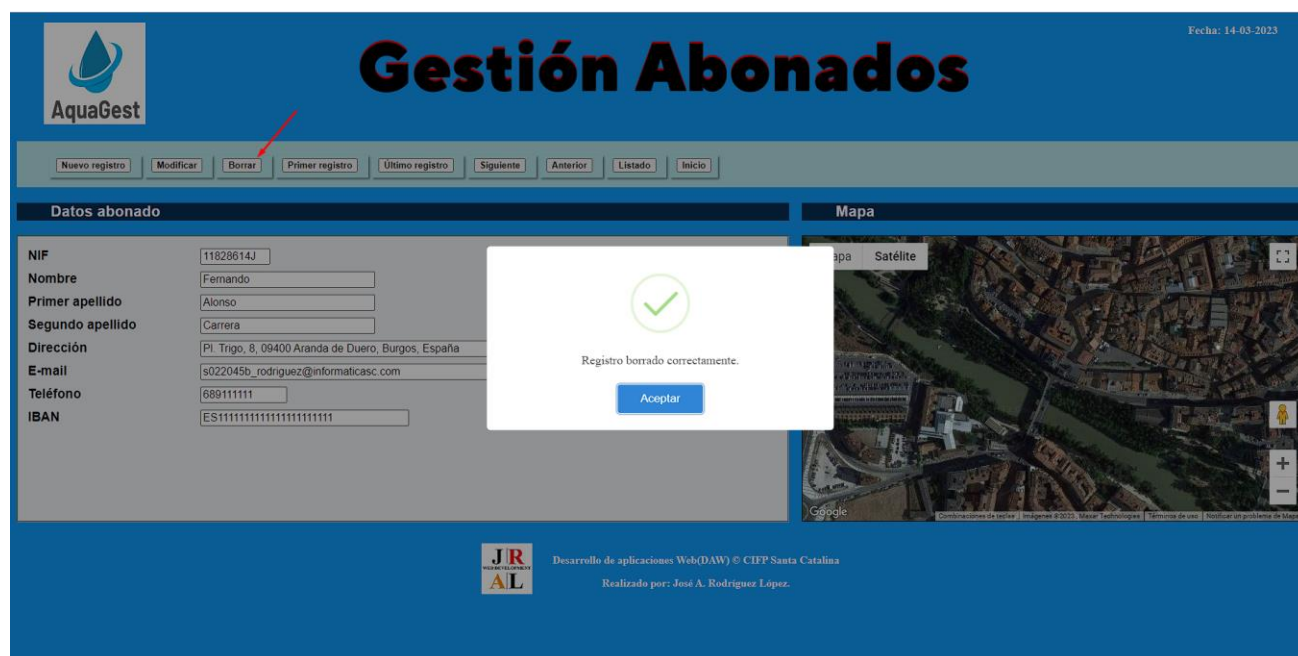
Registro modificado correctamente.

Aceptar

JR AL Desarrollo de aplicaciones Web(DAW) © CIEP Santa Catalina Realizado por: José A. Rodríguez López.

9. Borrar abonados:

Con esta opción se elimina un abonado de la base de datos. Por la definición de la base de datos, al eliminarlo se eliminan los registros de dispositivos y consumos asociados.



Fecha: 14-03-2023

Gestión Abonados

Nuevo registro Modificar **Borrar** Primer registro Último registro Siguiente Anterior Listado Inicio

Datos abonado

NIF: 11828614J
Nombre: Fernando
Primer apellido: Alonso
Segundo apellido: Carrera
Dirección: Pl. Trigo, 8, 09400 Aranda de Duero, Burgos, España
E-mail: s022045b_rodriguez@informaticasc.com
Teléfono: 689111111
IBAN: ES111111111111111111111111

Mapa

Registro borrado correctamente.

Aceptar

JR AL Desarrollo de aplicaciones Web(DAW) © CIEP Santa Catalina Realizado por: José A. Rodríguez López.

Con la opción Inicio del menú se va a la página principal.

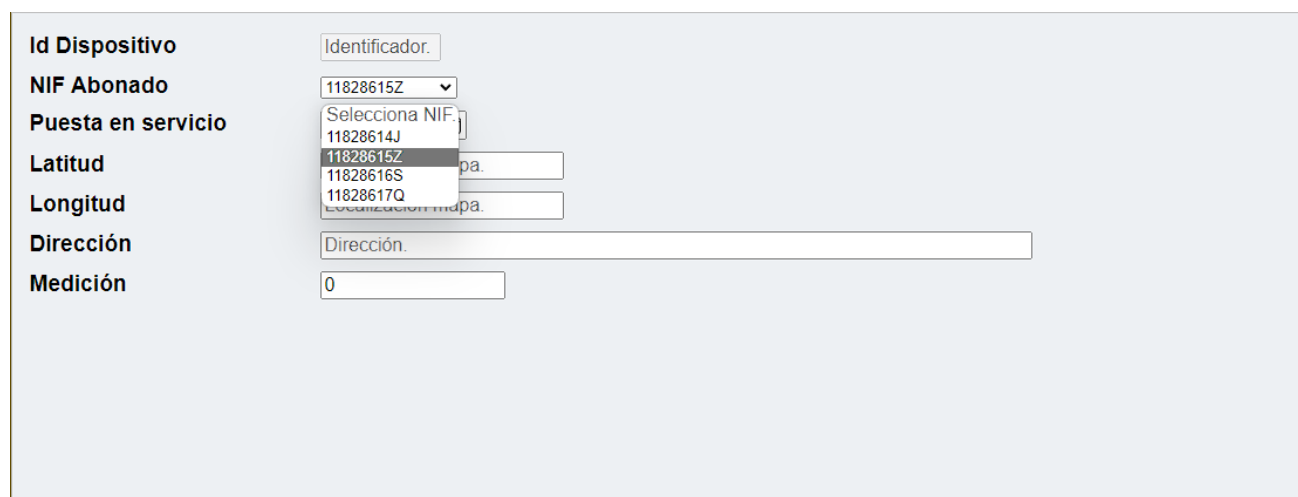
10. Creación de un nuevo dispositivo:

Pulsando sobre la opción de nuevo registro, se habilitará la grabación de este, pudiendo introducir la información solicitada.



The screenshot shows the AquaGest web application interface for 'Gestión Dispositivos'. The top header is blue with the AquaGest logo on the left and the date 'Fecha: 11-09-2023' on the right. Below the header is a navigation bar with buttons: 'Nuevo registro' (highlighted with a red arrow), 'Modificar', 'Borrar', 'Primer registro', 'Último registro', 'Siguiente', 'Anterior', 'Listado', 'Mapa Dispositivos', and 'Inicio'. The main content area is divided into two sections: 'Datos dispositivo' on the left and 'Mapa' on the right. The 'Datos dispositivo' section contains form fields for 'Id Dispositivo' (Identificador), 'NIF Abonado' (Selección NIF), 'Puesta en servicio' (dd/mm/aaaa), 'Latitud' (Localización mapa), 'Longitud' (Localización mapa), 'Dirección', and 'Medición' (0). The 'Mapa' section displays a Google Maps satellite view of a city area. At the bottom, there is a footer with logos for 'JR' and 'AL', and text indicating 'Desarrollo de aplicaciones Web(DAW) © CIFF Santa Catalina' and 'Realizado por: José A. Rodríguez López'.

El Id Dispositivo no estará habilitado, porque será asignado por el gestor de bases de datos al grabar el registro. En el NIF Abonado deberemos seleccionar el NIF de un abonado que ya se encuentre dado de alta en el sistema.



This screenshot shows the 'Datos dispositivo' form with the 'NIF Abonado' dropdown menu open. The dropdown list displays the following values: '11828615Z', '11828614J', '11828615Z', '11828616S', and '11828617Q'. The other form fields are visible but not interacted with: 'Id Dispositivo' (Identificador), 'Puesta en servicio' (dd/mm/aaaa), 'Latitud' (Localización mapa), 'Longitud' (Localización mapa), 'Dirección', and 'Medición' (0).

En la puesta en servicio admite la fecha no posterior a la actual del sistema (no confundir con la que se muestra en la interfaz con el fin de emular el funcionamiento) y no anterior al 1 de enero del 2020.


```
//Valida la puesta en servicio.
if ((evt.target.id === 'bGrabar' || evt.target.id === 'bModificar') && validado) {
    let patron = /^[0-9]{4}[-][0-9]{2}[-][0-9]{2}$/
    var resultado = patron.test(iPuestaServicio.value.trim())

    if (!resultado) {
        mensaje = 'La fecha introducida no es válida.'
        validado = false
    } else {
        //Procesa fecha para comprobar que es menor o igual a la actual.
        let fecha = iPuestaServicio.value.replaceAll('-', '')
        let fechaActual = obtenerFechaActual().replaceAll('-', '')
        if (fecha > fechaActual) {
            mensaje = 'La fecha no puede ser posterior a la actual.'
            validado = false
        } else if (fecha < 20200101) {
            mensaje = 'La fecha no puede ser anterior al 1 de enero del 2020.'
            validado = false
        }
    }
}
}
```

La latitud, longitud y dirección se asignarán haciendo click sobre el mapa en la localización del dispositivo ya que los campos no son editables.

La medición por defecto será 0, aunque puede ser modificada, pero no negativa. Solo admite un valor numérico.

```
//Valida la medida del dispositivo.
if ((evt.target.id === 'bGrabar' || evt.target.id === 'bModificar') && validado) {
    if (!iMedida.value) {
        mensaje = 'No se ha introducidos la medida del dispositivo.'
        validado = false
    } else if (iMedida.value < 0) {
        mensaje = 'La medida del dispositivo no puede ser negativa.'
        validado = false
    }
}
}
```



Gestión Dispositivos

Fecha: 27-01-2025

Nuevo registro

Modificar

Borrar

Primer registro

Último registro

Siguiente

Anterior

Listado

Mapa Dispositivos

Inicio

Datos dispositivo

Mapa

Id Dispositivo

23

NIF Abonado

11828615Z

Puesta en servicio

21/02/2023

Latitud

41.862322611744074

Longitud

-3.7074028394712655

Dirección

M76V+W2 Aranda de Duero, España

Medición

0


✓

Registro grabado correctamente.

Aceptar

Mapa

Satélite



Google

M76V+W2 Aranda de Duero, España

JR

AL


Desarrollo de aplicaciones Web(DAW) © CIEP Santa Catalina

Realizado por: José A. Rodríguez López.

El resto de las opciones de la interfaz tienen funciones similares a la de la interfaz de Abonados por lo que no se describe su funcionamiento.

11. Listado dispositivos:

Esta opción descarga un documento PDF con todos los dispositivos registrados en el sistema.



Listado Caudalímetros

ID	NIF Abonado	Puesta en servicio	Latitud	Longitud	Dirección	Medición
24	11828614J	2023-02-20	41.869405141888715	-3.6917857984630187	Av. Ruperta Baraya, 14, 09400 Aranda de Duero, Burgos, España	28.83
25	11828616S	2023-02-20	41.870530397701113	-3.68919662857333	Pl. Mayor, 25, 09400 Aranda de Duero, Burgos, España	30.54
26	11828614J	2023-02-20	41.866808181834575	-3.698931370916756	Variante I N-1, 1, 09400 Aranda de Duero, Burgos, España	29.96
27	11828617Q	2023-02-11	41.86820622947431	-3.692071956637295	C. Santo Domingo, 2, 09400 Aranda de Duero, Burgos, España	29.70
28	11828616S	2023-02-19	41.87350503044752	-3.6788003948298886	Cam. Alagunas, 4, 09400 Aranda de Duero, Burgos, España	28.58
29	11828615Z	2023-02-20	41.86126219407801	-3.697652209774307	M862 GW Aranda de Duero, España	29.80
30	11828615Z	2023-02-11	41.87150249500401	-3.675863351673827	C. Fernán González, 30, 09400 Aranda de Duero, Burgos, España	28.65
31	11828617Q	2023-02-20	41.872560204539586	-3.6820579861372327	C. Vela Zanetti, 8, 09400 Aranda de Duero, Burgos, España	28.36
32	11828615Z	2023-02-09	41.865550291903415	-3.6956882855635128	Calle Mz M6, 2, 09400 Aranda de Duero, Burgos, España	31.06
33	11828615Z	2023-02-11	41.86630819155316	-3.695596414076372	Manzana M5, 4, 09400 Aranda de Duero, Burgos, España	29.83
34	11828616S	2023-02-20	41.86436605213706	-3.6878036785625135	C. Burgo de Osma, 72, 09400 Aranda de Duero, Burgos, España	28.60

GESTION DE AGUAS-JOSE A. RODRIGUEZ LOPEZ

Pg: 1

JR

WEB DEVELOPMENT

AL

Proyecto de Gestión de Aguas Realizado por: José A. Rodríguez López

Pág. 20

12. Mapa dispositivos:

Muestra la localización de cada dispositivo en el mapa. Haciendo click sobre el marcador muestra la información del dispositivo.



13. Generador de consumos:

Con este fin, he creado una tarea periódica que cada cierto tiempo lee los dispositivos que están dados de alta en el sistema. Si hay dispositivos dados de alta en la base de datos, se genera un número aleatorio diferente

que se suma a la medida de cada uno de ellos y se actualiza en la base de datos. De esta manera consigo que las medidas sean diferentes para cada uno de ellos.

```
/*
Proyecto realizado por: José A. Rodríguez López
Fecha: 18/02/2023
*/

let intervaloTiempo = 10000 //Tiempo en milisegundos de la tarea programada.
5minuto.
let tareaGeneracionConsumo //Tarea temporizada que genera el consumo de los
caudalímetros aleatoriamente.

//-----
//Inicia la tarea periodica que genara los consumos de los dispositivos.
function inicioGeneracionConsumos() {
    //Ejecuta la función repetidamente cada intervalo indicado en milisegundos.
    tareaGeneracionConsumo = setInterval(() => {
        leerDispositivos(); //Lee el ID de los dispositivos
    }, intervaloTiempo)
}

//-----
//Función que genera los consumos aleatoriamente y los almacena en la base de datos.
function generarConsumo(dispositivos) {
    if (dispositivos) {
        for (let i = 0; i < dispositivos.length; i++) {
            //console.log(dispositivos[i].Id+"--
            +(parseFloat(dispositivos[i].Medida)+Math.random()*10/100))
            actualizarConsumo(dispositivos[i].Id,
            (parseFloat(dispositivos[i].Medida) + Math.random() * 10 / 100))
        }
    }
}

//-----
//Finalizar tarea periódica.
function finalizarTareaPeriodica(tarea) {
    clearInterval(tarea)
    iniciadoGeneradorFechas=false;
}

//-----
//Leer los registros de los dispositivos de la base de datos.
```

```

async function leerDispositivos() {
    //Proporciona una forma fácil de obtener información de una URL sin tener que
    recargar la página completa. XMLHttpRequest es ampliamente usado en la programación
    AJAX.
    //A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier
    tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y
    ftp).
    var ajaxrequest = new XMLHttpRequest()

    ajaxrequest.open(
        'POST',
        'https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/con
sultarDatosDispositivos.php',
        true,
    )
    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )
    ajaxrequest.onreadystatechange = async function () {
        //alert(ajaxrequest.readyState + "--" + ajaxrequest.status);
        if (ajaxrequest.readyState === 4 && ajaxrequest.status === 200) {
            let datosLeidos = ajaxrequest.responseText
            if (datosLeidos) {
                generarConsumo(JSON.parse(datosLeidos))
            }
        }
    }
    let envio = 'Envio'

    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )
    ajaxrequest.send(envio)
}

//-----
//Función que actualiza la medida de un dispositivo.
function actualizarConsumo(id, medida) {
    //Proporciona una forma fácil de obtener información de una URL sin tener que
    recargar la página completa. XMLHttpRequest es ampliamente usado en la programación
    AJAX.
    //A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier
    tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y
    ftp).
    let ajaxrequest = new XMLHttpRequest()

```

```

//Inicializa una solicitud recién creada o reinicializa una existente.
ajaxrequest.open(
    'POST',
    'https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/actualizaConsumo.php',
    true,
)

//Establece el valor encabezado de una solicitud HTTP. Al usarse, debe llamarse después de llamar a open(), pero antes de llamar a send().
//Si se llama a este método varias veces con el mismo encabezado, los valores se combinan en un único encabezado de solicitud.setRequestHeader()
ajaxrequest.setRequestHeader(
    'Content-type',
    'application/x-www-form-urlencoded',
)

//Cambio de estado a listo,
ajaxrequest.onreadystatechange = function () {
    //alert(ajaxrequest.readyState + '--' + ajaxrequest.status)
    if (ajaxrequest.readyState === 4 && ajaxrequest.status === 200) {
        let respuesta = ajaxrequest.responseText
        if (respuesta === "Error al actualizar el consumo.") {
            mostrarVentanaEmergente(respuesta, 'error')
        }
    }
}

ajaxrequest.setRequestHeader(
    'Content-type',
    'application/x-www-form-urlencoded',
)

//Envía la solicitud al servidor.
let envio = "Envio=" + id + "%*" + medida
ajaxrequest.send(envio)
}

//-----
//Inicia la tarea temporizada de generar caudales.
inicioGeneracionConsumos();

```


14. Generador de fechas:

Con fines de simulación, para emitir los correos de notificación con el importe a facturar sin tener que esperar a una fecha determinada, se crea una tarea periódica que funciona de la siguiente manera. Al inicio de la aplicación se lee la fecha más alta de facturación de los consumos que hay en la base de datos más 1 día y si no hay consumos se toma la fecha actual del sistema. La forma de proceder de esta manera está determinada porque la fecha de la medida en la tabla consumos actúa de clave primaria junto al identificador del dispositivo y daría error en caso de intentar grabar datos de consumo con la misma clave primaria.

Para poder mantener entre las diferentes páginas de la aplicación el valor del día, mes, año y una bandera de señalización para que no se vuelva a inicializar la fecha utilizo variables de sesión que voy modificando cada cierto intervalo (como es con fines de simulación se ha tomado que todos los meses tienen 30 días lo cual no supone ningún problema).

```
/*
Proyecto realizado por: José A. Rodríguez López
Fecha: 20/02/2023
*/

let tiempoGeneradorFechas = 2000 //Tiempo en milisegundos del generador de fechas
para el cambio de día.
let tareaGeneradorFechas //Tarea temporizada que genera fechas para la simulación.

//-----
//Busqueda de la fecha más alta en la base de datos y si no existe se coge la fecha
actual.
function buscarFechaMasAlta() {
    //Proporciona una forma fácil de obtener información de una URL sin tener que
    recargar la página completa. XMLHttpRequest es ampliamente usado en la programación
    AJAX.
    //A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier
    tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y
    ftp).
    var ajaxrequest = new XMLHttpRequest()

    ajaxrequest.open(
        'POST',
        'https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/fec
haMasAlta.php',
        true,
    )
    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
```

```

    )
    ajaxrequest.onreadystatechange = async function () {
        //alert(ajaxrequest.readyState + "--" + ajaxrequest.status);
        if (ajaxrequest.readyState === 4 && ajaxrequest.status === 200) {
            let datosLeidos = ajaxrequest.responseText
            fecha = JSON.parse(datosLeidos)[0].MaxFecha
            if (fecha) {
                procesarFechaMasAlta(fecha, 1)
            } else {
                procesarFechaMasAlta(obtenerFechaActual())
            }
        }
    }
    let envio = 'Envio'

    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )
    ajaxrequest.send(envio)
}

//-----
//Procesa la fecha para asignarlo a las variables globales.
function procesarFechaMasAlta(fecha, incremento = 0) {
    window.sessionStorage.setItem("dia", parseInt(fecha.substring(8)) + incremento)
    window.sessionStorage.setItem("mes", parseInt(fecha.substring(5, 7)))
    window.sessionStorage.setItem("anio", parseInt(fecha.substring(0, 4)))
    actualizarFecha();
}

//-----
//Inicia la tarea periodica que genera las fechas.
function inicioGeneradorFechas() {
    //Ejecuta la función repetidamente cada intervalo indicado en milisegundos.
    tareaGeneradorFechas = setInterval(() => {
        incrementaFecha()
    }, tiempoGeneradorFechas)
}

//-----
function incrementaFecha() {
    window.sessionStorage.setItem("dia",
    parseInt(window.sessionStorage.getItem("dia")) + 1)
}

```

```

        if (parseInt(window.sessionStorage.getItem("dia")) > 30) { //Para la
simulación se toma que todos los meses tienen 30 días.
            window.sessionStorage.setItem("dia", 1)
            window.sessionStorage.setItem("mes",
parseInt(window.sessionStorage.getItem("mes"))) + 1)
            if (parseInt(window.sessionStorage.getItem("mes")) > 12) {
                window.sessionStorage.setItem("mes", 1)
                window.sessionStorage.setItem("anio",
parseInt(window.sessionStorage.getItem("anio"))) + 1)
            }
        }
        actualizarFecha()
    }

//-----
//Obtiene la fecha de las variables globales.
function obtenerFecha(dia, mes, anio) {
    if (parseInt(dia) < 10) dia = '0' + dia //Agrega cero si es menor de 10
    if (parseInt(mes) < 10) mes = '0' + mes //Agrega cero si es menor de 10
    return anio + '-' + mes + '-' + dia
}

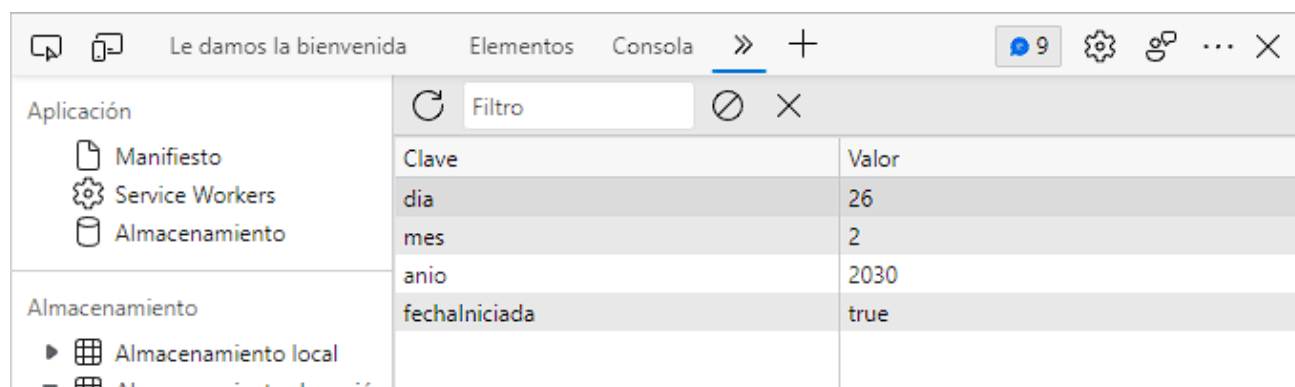
//-----
//Actualiza la fecha.
function actualizarFecha() {
    document.getElementById('fecha').innerHTML = "Fecha: " +
cambiarFormatoFecha(obtenerFecha(window.sessionStorage.getItem("dia"),
    window.sessionStorage.getItem("mes"),
window.sessionStorage.getItem("anio")))
}

//-----
//Inicia la fecha a mostrar en la interfaz.
if (!window.sessionStorage.getItem("fechaIniciada")) {
    buscarFechaMasAlta()
    actualizarFecha()
    window.sessionStorage.setItem("fechaIniciada", true)
} else {
    actualizarFecha();
}

//-----
//Ejecución.

```

```
inicioGeneradorFechas()
```



The screenshot shows a web application interface. On the left, there is a sidebar with a tree view containing 'Aplicación' (with sub-items 'Manifiesto', 'Service Workers', and 'Almacenamiento') and 'Almacenamiento' (with sub-items 'Almacenamiento local' and 'Almacenamiento de sesión'). The main area displays a table with a 'Filtro' input field and a table with two columns: 'Clave' and 'Valor'. The table contains the following data:

Clave	Valor
dia	26
mes	2
año	2030
fechaIniciada	true

15. Facturación:

La facturación se emite los días 30 de marzo, junio, septiembre y diciembre. Periódicamente se realiza la lectura de la medida de los dispositivos, si la fecha simulada coincide con alguna de esas fechas se activa el proceso de facturación grabando la medida del dispositivo y la fecha simulada en la tabla consumos junto al id del dispositivo y el precio del metro cubico. Tras la grabación por cada uno de los dispositivos se procede a la emisión de un correo electrónico con la información a cada uno de los abonados.

```
/*
Proyecto realizado por: José A. Rodríguez López
Fecha: 18/02/2023
*/

const PRECIO_METRO_CUBICO = 3.5 //Constante que define el importe facturado por cada
metro cubico de agua.
let facturado = false //Flag que indica si se ha producido la facturación.
let tiempoFacturacion = 1000 //Tiempo en milisegundos de la tarea programada
facturarConsumo.
let tareaFacturacion //Tarea temporizada que genera la facturación.

//-----
//Inicia la tarea periodica que recopila las medidas para la facturación.
function inicioFacturacionMedidas() {
    //Ejecuta la función repetidamente cada intervalo indicado en milisegundos.
    tareaFacturacion = setInterval(() => {
        leerMedidasDispositivos(); //Lee el ID de los dispositivos
    }, tiempoFacturacion)
}
```

```

//-----
//Leer los registros de los dispositivos de la base de datos.
async function leerMedidasDispositivos() {
    //Proporciona una forma fácil de obtener información de una URL sin tener que
    recargar la página completa. XMLHttpRequest es ampliamente usado en la programación
    AJAX.
    //A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier
    tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y
    ftp).
    var ajaxrequest = new XMLHttpRequest()

    ajaxrequest.open(
        'POST',
        'https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/con
sultarDatosDispositivos.php',
        true,
    )
    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )
    ajaxrequest.onreadystatechange = async function () {
        //alert(ajaxrequest.readyState + "--" + ajaxrequest.status);
        if (ajaxrequest.readyState === 4 && ajaxrequest.status === 200) {
            let datosLeidos = ajaxrequest.responseText
            if (datosLeidos) {
                facturacion(JSON.parse(datosLeidos))
            }
        }
    }
    let envio = 'Envio'

    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )
    ajaxrequest.send(envio)
}

//-----
//Recorre todos los dispositivos tomando sus medidas para guardarlos en la tabla de
consumos.
function facturacion(dispositivos) {
    //Si hay dispositivos y no ha sido ya facturado el consumo.
    if (dispositivos) {

```

```

        if ((window.sessionStorage.getItem("dia") == 30 &&
(window.sessionStorage.getItem("mes") == 3 || window.sessionStorage.getItem("mes")==
6 ||
        window.sessionStorage.getItem("mes") == 9 ||
window.sessionStorage.getItem("mes")== 12)) && !facturado) {
            for (let i = 0; i < dispositivos.length; i++) {
                grabarMedidas(dispositivos[i].Id,
obtenerFecha(window.sessionStorage.getItem("dia"),
window.sessionStorage.getItem("mes"),
                window.sessionStorage.getItem("anio")), dispositivos[i].Medida,
PRECIO_METRO_CUBICO)
            }
            facturado = true;
        } else {
            facturado = false
        }
    }
}

//-----
//Función que actualiza la medida de un dispositivo.
function grabarMedidas(IdDispositivo, fecha, medida, precio) {
    let consumo =
        '{' +
        'IdDispositivo' +
        ':' +
        '"' +
        IdDispositivo +
        "'," +
        '"' +
        'Fecha_medida' +
        ':' +
        '"' +
        fecha +
        "'," +
        '"' +
        'Medida' +
        ':' +
        '"' +
        medida +
        "'," +
        '"' +
        'Precio' +
        ':' +
        '"' +
        precio +
        '}'

```

//Proporciona una forma fácil de obtener información de una URL sin tener que recargar la página completa. XMLHttpRequest es ampliamente usado en la programación AJAX.

//A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y ftp).

```
let ajaxrequest = new XMLHttpRequest()
```

//Inicializa una solicitud recién creada o reinicializa una existente.

```
ajaxrequest.open(  
    'POST',  
    'https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/grabarConsumos.php',  
    true,  
)
```

//Establece el valor encabezado de una solicitud HTTP. Al usarse, debe llamarse después de llamar a open(), pero antes de llamar a send().

//Si se llama a este método varias veces con el mismo encabezado, los valores se combinan en un único encabezado de solicitud.setRequestHeader()

```
ajaxrequest.setRequestHeader(  
    'Content-type',  
    'application/x-www-form-urlencoded',  
)
```

//Cambio de estado a listo,

```
ajaxrequest.onreadystatechange = function () {  
    //alert(ajaxrequest.readyState + '--' + ajaxrequest.status)  
    if (ajaxrequest.readyState === 4 && ajaxrequest.status === 200) {  
        let respuesta = ajaxrequest.responseText;  
        if (respuesta === "Registro grabado correctamente.") {  
            enviarCorreoConsumo(IdDispositivo)  
        } else {  
            mostrarVentanaEmergente(respuesta, 'error')  
        }  
    }  
}
```

```
ajaxrequest.setRequestHeader(  
    'Content-type',  
    'application/x-www-form-urlencoded',  
)
```

//Envía la solicitud al servidor.

```
let envio = 'Todo=' + consumo  
ajaxrequest.send(envio)
```

```
}
```

```

//-----
//Envia el correo de consumo al abonado.
function enviarCorreoConsumo(IdDispositivo) {
    //Proporciona una forma fácil de obtener información de una URL sin tener que
    recargar la página completa. XMLHttpRequest es ampliamente usado en la programación
    AJAX.
    //A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier
    tipo de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y
    ftp).
    let ajaxrequest = new XMLHttpRequest()

    //Inicializa una solicitud recién creada o reinicializa una existente.
    ajaxrequest.open(
        'POST',
        'https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/con
sultarDatosCorreo.php',
        true,
    )

    //Establece el valor encabezado de una solicitud HTTP. Al usarse, debe llamarse
    después de llamar a open(), pero antes de llamar a send().
    //Si se llama a este método varias veces con el mismo encabezado, los valores se
    combinan en un único encabezado de solicitud.setRequestHeader()
    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )

    //Cambio de estado a listo,
    ajaxrequest.onreadystatechange = function () {
        //alert(ajaxrequest.readyState + '--' + ajaxrequest.status)
        if (ajaxrequest.readyState === 4 && ajaxrequest.status === 200) {
            let datosLeidos = ajaxrequest.responseText;
            if (datosLeidos) {
                setTimeout(() => {procesaCorreo(datosLeidos)}, 5);
            } else {
                mostrarVentanaEmergente('Error en la lectura de datos para el envío
del correo.', 'error')
            }
        }
    }

    ajaxrequest.setRequestHeader(
        'Content-type',
        'application/x-www-form-urlencoded',
    )
}

```



```

//Envía la solicitud al servidor.
let envio = 'Envio=' + IdDispositivo
ajaxrequest.send(envio)
}

//-----
//Procesa los datos y realiza el envío del correo con la cantidad a facturar.
function procesaCorreo(datosLeidos) {
    let datos = JSON.parse(datosLeidos)
    let consumo = 0;
    if (datos.length == 1) {
        consumo = datos[0].Medida;
    } else {
        consumo = datos[0].Medida - datos[1].Medida;
    }

    //Datos del correo electrónico.
    let datosCorreo =
        '{' +
        'Titulo' +
        ':' +
        '"Información del contrato del suministro de aguas",' +
        'NIF' +
        ':' +
        'datos[0].NIF +
        ',' +
        'Nombre' +
        ':' +
        'datos[0].Nombre + " " + datos[0].Apellido1 + " " + datos[0].Apellido2 +
        ',' +
        'Email' +
        ':' +
        'datos[0].Email +
        ',' +
        'Direccion' +
        ':' +
        'datos[0].Direccion +
        ',' +

```

```

        '"' +
        'Fecha' +
        ':' +
        '"' +
        cambiarFormatoFecha(datos[0].Fecha_medida) +
        ',' +
        '"' +
        'Importe' +
        ':' +
        '"' +
        Math.round(consumo * datos[0].Precio * 100) / 100 +
        '"}'

console.log(datosCorreo)

enviarEmail(datosCorreo)
}

//-----
//Inicio de la facturación.
inicioFacturacionMedidas()

```

✓ Mostrando filas 0 - 24 (total de 187, La consulta tardó 0.0005 segundos.) [Fecha_medida: 2030-0

SELECT * FROM `consumos` ORDER BY `consumos`.`Fecha_medida` DESC

1 > >> | ☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

		IdDispositivo	Fecha_medida	Medida	Precio
<input type="checkbox"/>	Editar Copiar Borrar	27	2030-06-30	44.14	3.50
<input type="checkbox"/>	Editar Copiar Borrar	33	2030-06-30	44.92	3.50
<input type="checkbox"/>	Editar Copiar Borrar	28	2030-06-30	43.62	3.50
<input type="checkbox"/>	Editar Copiar Borrar	32	2030-06-30	46.49	3.50
<input type="checkbox"/>	Editar Copiar Borrar	29	2030-06-30	44.31	3.50
<input type="checkbox"/>	Editar Copiar Borrar	25	2030-06-30	46.02	3.50
<input type="checkbox"/>	Editar Copiar Borrar	26	2030-06-30	45.25	3.50
<input type="checkbox"/>	Editar Copiar Borrar	34	2030-06-30	43.27	3.50
<input type="checkbox"/>	Editar Copiar Borrar	31	2030-06-30	43.74	3.50
<input type="checkbox"/>	Editar Copiar Borrar	30	2030-06-30	43.92	3.50
<input type="checkbox"/>	Editar Copiar Borrar	24	2030-06-30	44.20	3.50
<input type="checkbox"/>	Editar Copiar Borrar	27	2030-03-30	43.39	3.50
<input type="checkbox"/>	Editar Copiar Borrar	26	2030-03-30	44.39	3.50
<input type="checkbox"/>	Editar Copiar Borrar	29	2030-03-30	43.34	3.50
<input type="checkbox"/>	Editar Copiar Borrar	24	2030-03-30	43.23	3.50
<input type="checkbox"/>	Editar Copiar Borrar	32	2030-03-30	45.43	3.50
<input type="checkbox"/>	Editar Copiar Borrar	33	2030-03-30	43.88	3.50
<input type="checkbox"/>	Editar Copiar Borrar	30	2030-03-30	42.94	3.50
<input type="checkbox"/>	Editar Copiar Borrar	28	2030-03-30	42.63	3.50
<input type="checkbox"/>	Editar Copiar Borrar	31	2030-03-30	42.86	3.50
<input type="checkbox"/>	Editar Copiar Borrar	25	2030-03-30	45.02	3.50
<input type="checkbox"/>	Editar Copiar Borrar	34	2030-03-30	42.33	3.50
<input type="checkbox"/>	Editar Copiar Borrar	24	2029-12-30	42.38	3.50

Consola

16. Emisión correo electrónico:

Tras la preparación de los datos en el proceso de facturación se llama la función de enviarEmail. Esta función utiliza el archivo PHP envio_email.php que será el encargado de ejecutarse en el servidor para el envío del correo electrónico.

```
//-----  
-----  
//Función que envía un email  
function enviarEmail(datosCorreo) {  
  
    //Proporciona una forma fácil de obtener información de una URL sin tener que  
    recargar la página completa. XMLHttpRequest es ampliamente usado en la programación  
    AJAX.  
    //A pesar de su nombre, XMLHttpRequest puede ser usado para recibir cualquier tipo  
    de dato, no solo XML, y admite otros formatos además de HTTP (incluyendo file y  
    ftp).  
    let ajaxrequest = new XMLHttpRequest()  
  
    //Inicializa una solicitud recién creada o reinicializa una existente.  
    ajaxrequest.open("POST",  
"https://www.informaticasc.com/curso22_23/Rodriguez/Gestion_de_aguas/php/envio_email  
.php", true);  
  
    //Cambio de estado a listo,  
    ajaxrequest.onreadystatechange = function () {  
        if (ajaxrequest.readyState == 4) {  
            respuesta = ajaxrequest.responseText  
            if(respuesta!=""){  
                mostrarVentanaEmergente(respuesta,"error")  
            }  
        }  
    }  
}  
  
    ajaxrequest.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");  
  
    //Envía la solicitud al servidor.  
    let envio = "Envio="+datosCorreo  
    ajaxrequest.send(envio)  
}
```

```
<?php  
  
require('PHPMailer.php');  
  
header('Access-Control-Allow-Origin: *');  
header('Access-Control-Allow-Headers: Content-Type');
```

```

//Recibe el array con los datos JSON.
$contenido = $_POST['Envio'];
//Acondicionamiento de la cadena para ser tratada.
$contenido = str_replace("\\", "", $contenido);
//Se decodifican los datos JSON.
$array = json_decode($contenido, true);
//Extracción de los datos.
$tmpArray = array();

foreach ($array as $dato) {
    $tmpArray[] = $dato;
}

$mail = new PHPMailer();
try {

    //Destinatario
    $mail->setFrom('gestionaguasaranda@gmail.com','SERVICIO_AGUAS');
    $mail->addAddress($tmpArray[3]); //Añade un destinatario
    $mail->addReplyTo('gestionaguasaranda@gmail.com');

    //Content
    $mail->isHTML(true);
    $mail->Subject = utf8_decode($tmpArray[0]);
    $mail->Body = utf8_decode("<p><b>Fecha: </b>$tmpArray[5]</p><p><b>Dirección:
</b>$tmpArray[4]</p>".
    "<p><b>Hola $tmpArray[2]:</b></p><p>En los próximos días, procederemos a
domiciliar el recibo en su cuenta
por el servicio prestado con un importe de $tmpArray[6]&euro;.</p><p>Los
servicios municipales de aguas le
saludan atentamente.</p>");


    //Attachments
    $mail->AddAttachment("ficheroAEnviar.pdf"); //Opcional

    $mail->send();
} catch (Exception $e) {
    echo "Error en el envío. Mailer Error: {$mail->ErrorInfo}";
}

```

Con el fin de poder realizar el envío de un correo electrónico creé una cuenta Gmail para tal efecto gestionaguasaranda@gmail.com Para esta cuenta he tenido que activar la verificación en dos pasos para poder ser utilizada y crear una contraseña de aplicación.

Iniciar sesión en Google



Contraseña	Última modificación: 19 feb	>
Verificación en dos pasos	<input checked="" type="checkbox"/> Activada	>
Contraseñas de aplicaciones	1 contraseña	>

← Contraseñas de aplicaciones

Las contraseñas de aplicación te permiten iniciar sesión en tu cuenta de Google desde aplicaciones instaladas en dispositivos que no admiten la verificación en dos pasos. No tendrás que recordarlas porque solo tienes que introducirlas una vez. [Más información](#)

Tus contraseñas de aplicación

Nombre	Fecha de creación	Último uso
Gestion_de_aguas	20 feb	–

Selecciona la aplicación y el dispositivo para los que quieres generar la contraseña de aplicación.

Seleccionar aplicación

▼

Seleccionar dispositivo

▼

GENERAR

Ejemplo del correo emitido:

Información del contrato del suministro de aguas



SERVICIO_AGUAS <gestionaguasaranda@gmail.com>

19:54



Para: j062667@gmail.com

Fecha: 30-03-2031

Dirección: C. Santo Domingo, 2, 09400 Aranda de Duero, Burgos, España

Hola Pedro Sanchez Arnaiz:

En los próximos días, procederemos a domiciliar el recibo en su cuenta por el servicio prestado con un importe de 3.05€.

Los servicios municipales de aguas le saludan atentamente.

En el servidor se han creado varias cuentas con fines de depuración, pues si múltiples correos se envían a la misma dirección e-mail a la vez el servidor lo toma como un ataque y los bloquea.

<input type="checkbox"/>	>	s022045b_rodriguez@inform...	✓ Ilimitado	220,23 KB / 250 MB / 0,09%	CHECK EMAIL	ADMINISTRAR	CONNECT DEVICES
<input type="checkbox"/>	>	s022045b_rodriguez1@infor...	✓ Ilimitado	369,36 KB / 250 MB / 0,14%	CHECK EMAIL	ADMINISTRAR	CONNECT DEVICES
<input type="checkbox"/>	>	s022045b_rodriguez2@infor...	✓ Ilimitado	292,13 KB / 250 MB / 0,11%	CHECK EMAIL	ADMINISTRAR	CONNECT DEVICES

17. Despliegue de la aplicación en masbaratoimposible.com

Se han subido los archivos de la aplicación al servidor mediante la aplicación FileZilla a mi carpeta.

Se crea subdominio:

jarl.informaticasc.com

[/public_html/curso22_23/Rodriguez/Gestion_de_aguas](#)

Se asigna redireccionamiento:

[/public_html/curso22_23/Rodriguez/Gestion_de_aguas](#)

<http://jarl.informaticasc.com/index.php>

Se habilita la ejecución de código php.

jarl.informaticasc.com

PHP 7.4.6

Desactivar

Activar PHP 5.6.40
Activar PHP 7.0.33
Activar PHP 7.1.33
Activar PHP 7.2.30
Activar PHP 7.3.17
Activar PHP 7.4.6
Activar PHP 5.2.17
Activar PHP 5.3.29
Activar PHP 5.4.45
Activar PHP 5.5.38



Gestión de aguas

Fecha: 27-02-2023



Desarrollo de aplicaciones Web(DAW) © CHP Santa Catalina
Realizado por: José A. Rodríguez López.