

pr200 Documentación técnica

@Moises Muñoz

Índice:

Requisitos funcionales

Requisitos no funcionales

Asignación: Moisés, Z400, S2, S3

Prototipo recopilador datos + Almacenaje

Api recepción datos Arduino e inserción en base de datos

Devolución API recepción a Monitor serie.

API dar datos

Devolución API

Api dar urls para datos prueba

Api dar urls datos colaborativos

Diseño Aplicación

Aplicación Web

Gestión solicitudes fallidas

Promesas y uso de API interna urls

Tests

Resumen usuario jca

Usuario jca servidor y su llave pública

Usuario jca mysql y permisos base de datos

Requisitos funcionales

- ✓ ~~Prototipo obtención datos.~~
- ✓ ~~API~~
- ✓ ~~Aplicación Web~~

Requisitos no funcionales

- ✓ ~~Orientación a objetos~~
- ✓ ~~RWD~~

- ✓ Batería de pruebas automatizadas
- ✓ Modelado y diseño de base de datos
- ✓ Presentación y maquetado profesional
- ✓ Uso de herramientas de control de versiones para el desarrollo del proyecto (desde su inicio)
- ✓ Publicación y puesta en producción en el subdominio `pr200.newflow.tech`
- ✓ La aplicación debe estar desplegada y operativa en el servidor de **aws academy**.
- ✓ Estará debidamente documentado con bloques de documentación entendibles por `phpDocumentor`
- ✓ La documentación del código estará generada y desplegada en producción en la carpeta **02-docphp**
- ✓ Uso de contenedores **docker** para todos los procesos que se necesiten, debidamente documentados y explicados.
- ✓ Crear el usuario en el servidor **jca** con contraseña **daw** y que pertenezca al grupo **sudo**.
- ✓ Cargar la clave pública "**jca_daw_rsa.pub**" suministrada [aquí](#) en el `~/.ssh/authorized_keys` del usuario **jca**.
- ✓ Crear el usuario **jca** con acceso a `mysql` y permisos para poder trabajar en la **base de datos** donde está el proyecto.
- ✓ Documentación (en el directorio **01-doc**)
 - ✓ Documentación técnica:
 - ✓ checklist en la primera página con todos los requisitos
 - ✓ solución se ha aportado y cómo se ha resuelto
 - ✓ modelos conceptual, lógico y físico y su debida justificación (imágenes de diagramas)
 - ✓ Diagramas de Clases del proyecto
 - ✓ Justificación del diseño de la aplicación web
 - ✓ Manual de usuario:
 - ✓ Cómo utilizar cada uno de los servicios y que quede bien explicado.
 - ✓ Detalle del dominio para poder acceder al servidor y contraseñas necesarias.
 - ✓ Manual de instalación:
 - ✓ Requisitos de sistema

Asignación: Moisés, Z400, S2, S3

- S2) registro de entrada a algún área Bxxx
- S3) cantidad de luz que incide en la zona (Zxxx), tomando 1 muestra cada minuto

S2) mediante un lector de RFID

S3) haciendo uso de un sensor de luz

Prototipo recopilador datos + Almacenaje

Consistiendo en el desarrollo electrónico funcional que sea capaz de recabar la información que se solicita y que use un servicio que será encargado de almacenar la siguiente información:

- S2) hora en la que se realizó la entrada y área
- S3) hora en la que se realizó la muestra, área y cantidad de luz
- Dentro de la aplicación en app/arc/codeArduino se encuentra el programa que ejecuta el prototipo de Arduino, adjunto hoja con la configuración de placa, librerías y de los sensores necesarios.
- destaco las partes de envío de información del programa

Envío del valor de variable postData mediante POST a la API del servidor /clientdatos.php

```
void loop(){  
  
  if( ingresarEnBd == 1 ){  
    client.println("POST /api/clientdatos.php HTTP/1.1");  
    client.println("Host: pr200.newflow.tech");  
    client.println("Content-Type: application/x-www-form-urlencoded");
```

```

client.print("Content-Length: ");
client.println(postData.length());
client.println();
client.println(postData);
ingresarEnBd = 0;
}

```

Lectura de luminosidad cada minuto, mapeo de datos, asignación valor postData, variable ingresarEnBd = 1 para comunicarnos con el servidor.

```

if( millis() > tiempoActual + 25000 ){ // cada minuto lectura de luminosidad
    tiempoActual = millis();
    Serial.println( analogRead(A0) );
    postVariable = "iluminacion=";
    float magnitudValor = map(analogRead(A0), valorMinSensor, valorMaxSensor, 100, 0);
    postData = postVariable + magnitudValor;
    ingresarEnBd = 1;
}

```

Existe función verificadora de la llave, (chip) que se comunica con tarjeta RFID, de la misma manera que en anterior código asignación de postData e ingresarEnBd = 1 , para ingreso en base de datos del servidor.

```

if( verificaUID( LecturaUID, llave ) ){ // Verificación de UID
    Serial.println(" Acceso permitido");
    postVariable = "entradas=";
    int magnitudValor = 1;
    postData = postVariable + magnitudValor;
    ingresarEnBd = 1;
}
else {
    Serial.println(" Acceso NO permitido");
}
}

```

Api recepción datos Arduino e inserción en base de datos

La API se define en una clase llamada ApiReceive.php

- El cliente Arduino se conectará mediante esta url base:
<https://pr200.newflow.tech/api/clientdatos.php>

```

<?php
require '../vendor/autoload.php';
use Moi\Zonas\ApiReceive;

echo " Api Receive <br> ";
$apireceive = new ApiReceive();
/**
 * El registro de entrada se almacenada en servidor
 * El servicio devolverá al cliente si ha sido insertada con éxito
 */
if( isset( $_POST['entradas'] ) ){
    $apireceive -> entradas( $_POST['entradas'] );
}
/**
 * El registro de iluminación se almacenada en servidor
 * El servicio devolverá al cliente si ha sido insertada con éxito
 */
if( isset( $_POST['iluminacion'] ) ){
    $apireceive -> iluminacion( $_POST['iluminacion'] );
}

```

Devolución API recepción a Monitor serie.

<https://pr200.newflow.tech/api/apiReceive.php>

- Verifico desde cliente Arduino la respuesta de la API que guarda los datos.
 - Datos de luminosidad cada minuto y medio, fecha y hora guardas en el momento del registro en la base de datos.

```

HTTP/1.1 200 OK
Date: Mon, 06 Dec 2021 21:56:01 GMT
Server: Apache/2.4.51 (Unix) OpenSSL/1.1.1l
X-Powered-By: PHP/8.1.0RC5
Strict-Transport-Security: max-age=63072000
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

4b
  Api Receive <br> Recibido dato Luminosidad : 84.00 Iluminacion registrada
0

859
HTTP/1.1 200 OK
Date: Mon, 06 Dec 2021 21:56:26 GMT
Server: Apache/2.4.51 (Unix) OpenSSL/1.1.1l
X-Powered-By: PHP/8.1.0RC5
Strict-Transport-Security: max-age=63072000
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

4b
  Api Receive <br> Recibido dato Luminosidad : 86.00 Iluminacion registrada
0

```

- Datos de entrada mediante dispositivo RFID, fecha y hora guardas en el momento del registro en la base de datos.

```

UID leido : EC 8F 44 4A Acceso permitido
HTTP/1.1 200 OK
Date: Mon, 06 Dec 2021 21:59:52 GMT
Server: Apache/2.4.51 (Unix) OpenSSL/1.1.1l
X-Powered-By: PHP/8.1.0RC5
Strict-Transport-Security: max-age=63072000
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

44
  Api Receive <br> Recibida entrada RFID : 1 Entrada RFID registrada
0

```

- Registro de entradas luminosidad visto en monitor, base datos en container en servidor web AWSlab

```

Content-Type: text/html; charset=UTF-8

4b
  Api Receive <br> Recibido dato Luminosidad : 90.00 Iluminacion registrada
0

```

41	2021-12-06 22:00:11	87
42	2021-12-06 22:00:36	87
43	2021-12-06 22:01:01	87
44	2021-12-06 22:01:26	87
45	2021-12-06 22:30:14	90

45 rows in set (0.00 sec)

```

Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8

```

```

b
Api Receive <br> Recibido dato Luminosidad : 89.00 Iluminacion registrada

```

40	2021-12-06 21:59:46	86
41	2021-12-06 22:00:11	87
42	2021-12-06 22:00:36	87
43	2021-12-06 22:01:01	87
44	2021-12-06 22:01:26	87
45	2021-12-06 22:30:14	90
46	2021-12-06 22:30:39	89

46 rows in set (0.01 sec)

- Registro de entrada RFID visto en monitor, base datos en container en servidor web AWSLab

```

14
Api Receive <br> Recibida entrada RFID : 1 Entrada RFID registrada
)

```

37	2021-12-06 21:50:59	0x01
38	2021-12-06 21:50:59	0x01
39	2021-12-06 21:59:52	0x01
40	2021-12-06 22:31:41	0x01

40 rows in set (0.00 sec)

API dar datos

Preparar el servicio adecuado para poder proporcionar la información recabada. La información **tendrá la misma estructura**, para el mismo tipo de información (JSON).

La API se define en una clase llamada Api.php, donde se recopila la información deseada y se devuelve como JSON.

- El cliente Compañeros en este proyecto colaborativo podrán solicitar los datos mediante esta url base: <https://pr200.newflow.tech/api/moiz400.php>
- La url base de esta API cambio a <https://pr200.newflow.tech/api/api.php>

Devolución API

- desde servidor web AWSLab



The screenshot shows a web browser window with the address bar displaying `pr200.newflow.tech/api/moiz400.php`. The browser's developer tools are open, showing the response of an API call. The response is a JSON object with a key `"mediciones"` containing an array of three objects. Each object has the following fields: `"nombreZona"`, `"nombreArea"`, `"fecha"`, `"magnitud"`, and `"valor"`.

```
1 // 20211206233428
2 // https://pr200.newflow.tech/api/moiz400.php
3
4 {
5   "mediciones": [
6     {
7       "nombreZona": "Z400",
8       "nombreArea": null,
9       "fecha": "2021-12-06 21:50:59",
10      "magnitud": "iluminacion",
11      "valor": "41"
12    },
13    {
14      "nombreZona": "Z400",
15      "nombreArea": "B401",
16      "fecha": "2021-12-06 21:52:21",
17      "magnitud": "entradas",
18      "valor": "38"
19    },
20    {
21      "nombreZona": "Z400",
22      "nombreArea": "B402",
23      "fecha": "2021-12-06 21:52:21",
24      "magnitud": "entradas",
25      "valor": "38"
26    }
27  ]
28 }
```

Api dar urls para datos prueba

Devolución de JSON para ser tratado mediante AJAX en la parte cliente de la aplicación.

<https://pr200.newflow.tech/api/urlsPrueba.php>


```
[
  "https://newflow.tech/pr-200-jsonPruebas/z100-1.php",
  "https://newflow.tech/pr-200-jsonPruebas/z100-2.php",
  "https://newflow.tech/pr-200-jsonPruebas/z200-1.php",
  "https://newflow.tech/pr-200-jsonPruebas/z200-2.php",
  "https://newflow.tech/pr-200-jsonPruebas/z300-1.php",
  "https://newflow.tech/pr-200-jsonPruebas/z300-2.php",
  "https://newflow.tech/pr-200-jsonPruebas/z400-1.php",
  "https://newflow.tech/pr-200-jsonPruebas/z400-2.php",
  "https://newflow.tech/pr-200-jsonPruebas/z500-1.php",
  "https://newflow.tech/pr-200-jsonPruebas/z500-2.php"
]
```

Api dar urls datos colaborativos

Devolución de JSON para ser tratado mediante AJAX en la parte cliente de la aplicación.

<https://pr200.newflow.tech/api/urlsColab.php>

```
[
  "https://newflow.tech/pr-200-jsonPruebas/z500-2.php",
  "http://arduino.proyectos-andres.tech/zonas/pr200/",
  "http://samuel080.me",
  "http://pr200.rafadaw.software/app/mediciones/",
  "https://pr200.jose-velazquez-daw.rocks/api.php",
  "manu.....",
  "ballesteros.....",
  "pablo.....",
  "https://pr200.newflow.tech/api/api.php",
  "http://ismaeldaw080.me/apiIsmael.php",
  "ocete....."
]
```

Diseño Aplicación

El diseño será sencillo y esencialista,

- HTML

Dispone de dos templates usados para mostrar los datos obtenidos de los servicios

```
<template id="templatezonaSensor">
  <div id="divz">
    <fecha id="fechaz"> </fecha>
    <magnitud id="magnitudz"> </magnitud>
    <valor id="valorz"> </valor>
  </div>
</template>
<template id="templateareaSensor">
  <h3 id="areaa"> </h3>
  <div id="diva">
    <fecha id="fechaa"> </fecha>
    <magnitud id="magnituda"> </magnitud>
    <valor id="valora"> </valor>
  </div>
</template>
```

Una cabecera donde podremos acceder a las distintas partes.

Una parte principal donde un artículo recogerá los datos de la Zona dentro de una sección y los datos de las áreas.

```

<header class="header">
  <h1> Zonas de Almacenamiento </h1>
  <nav class="headernav">
    <ul>
      <li> <a id="datequipo" href="index.php">Datos equipo </a> </li>
      <li> <a id="datprueba" href="prueba.php">Datos prueba </a> </li>
    </ul>
  </nav>
</header>
<main class="main">
  <article class="article" id="article1">
    <section class="section" id="section1">
      <h1 id="h1"> Z100 </h1>
      <!--  -->
    </section>
  </article>

```

Un pie de pagina sencillo con acceso a documentación sobre las APIs

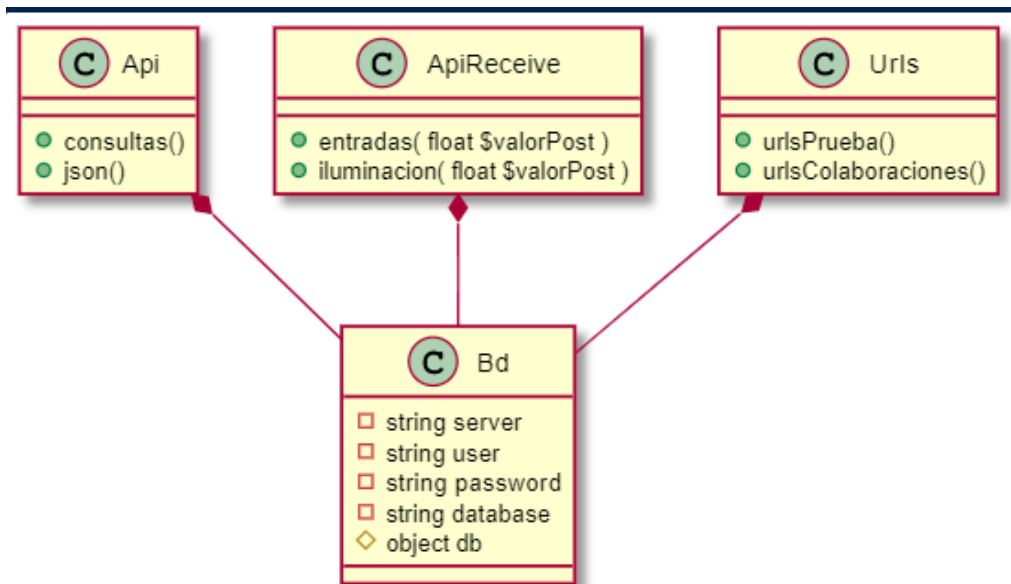
```

<footer class="footer">
  <section class="mercadosMateriasPrimas">
    <a href="#" class="imgfoot"> <img src="" alt="" /> </a>
  </section>
  <a href="api/docs/docapi.html">API</a>
</footer>

<script src="js/script1.js" language="javascript" type="text/javascript"></script>
<script src="js/script3.js" language="javascript" type="text/javascript"></script>

```

- Lógica usada en las API



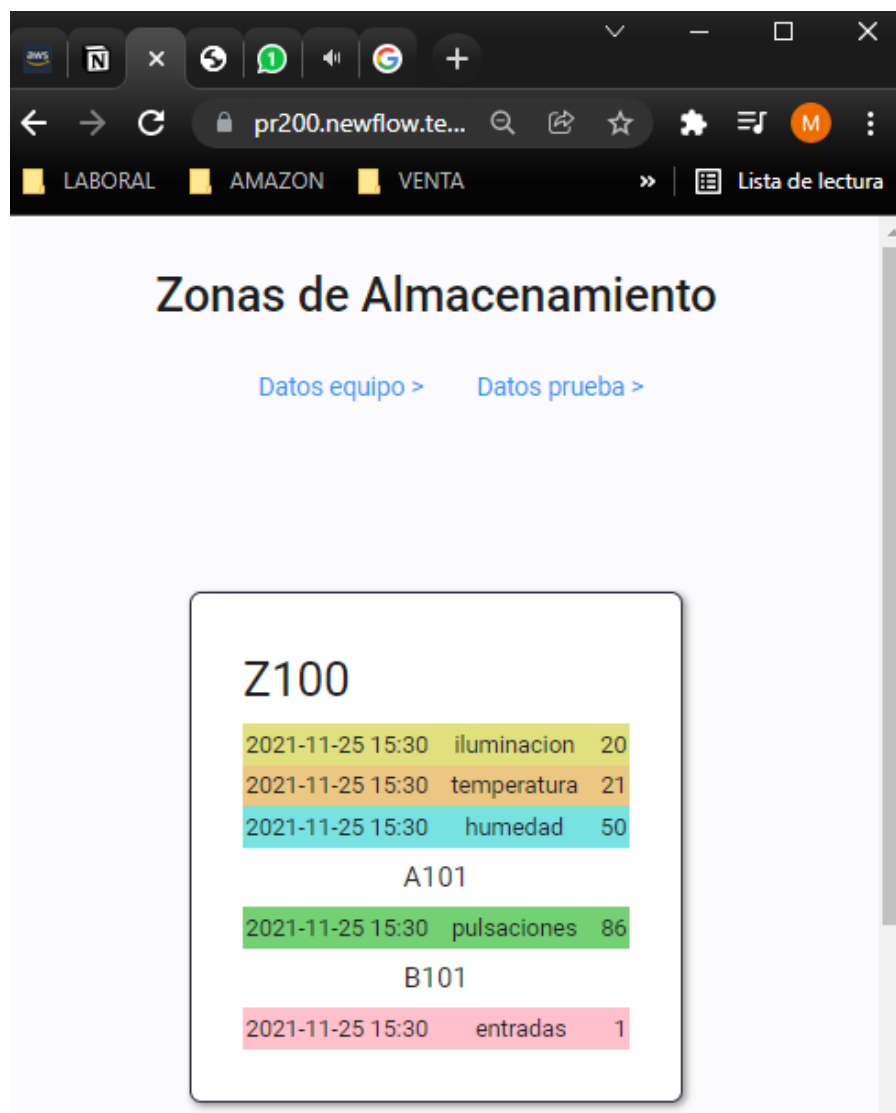
- CSS

Los estilos usados son sencillos, predomina la información y la identificación de la misma mediante uso de colores.

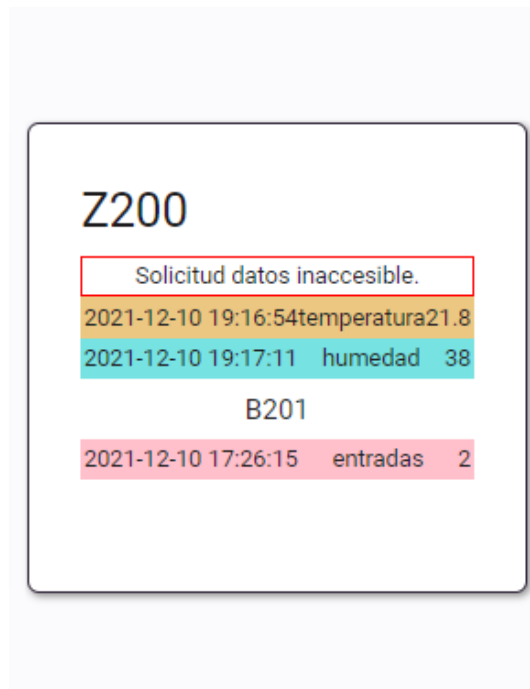
Se utiliza flexbox para maquetar la web.

RWD mediante media queries.

Los blancos no son blancos puros, ni los negros negros puros.



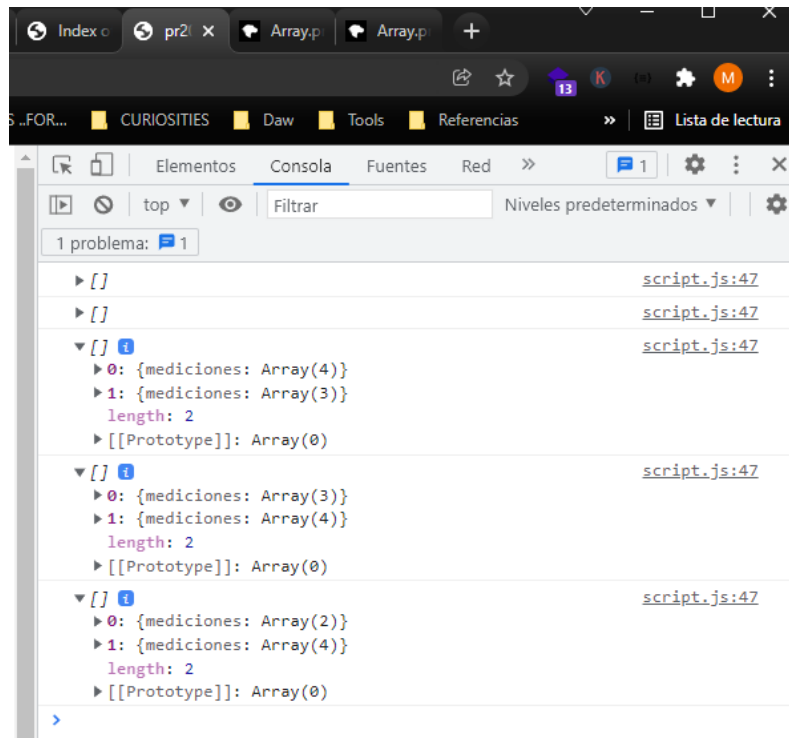
La gestión de los erros de solicitud de información son tratados, informando al cliente agradablemente.



Aplicación Web

Aplicación que consume **TODOS** los servicios de **TODOS** los trabajadores, se muestran todas las zonas y todos los datos, de forma agradable.

- Prueba consumiendo servicios de servidor remoto, desde mi aplicación, emulación de servicios reales a recibir.



- Genero código Javascript para modelar los datos esperados en la diversas peticiones a los compañeros del proyecto, la recepción será de manera asíncrona y la trataré con promesas...

```
function pintasolicitudes( solicitudes, article ){

  for( let i = 0; i < 2; i++ ){
    promise = fetch(`${solicitudes[i]}`);
    promise
    .then( response => {
      return response.json();
    })
    .then( data => {
      let clon ="clon";

      data.mediciones.forEach( (medicion,index) => {
        // console.log(medicion);
        if( medicion.nombreArea == null ){
          this[clon+solicitudes[0]+index] = templatezonaSensor.content.cloneNode(true);
        }
      })
    })
  }
}
```

El código comentado se puede encontrar en `app/src/js/script1.js`

- Pensé en contextualizar cada zona como si esta fuera un silo o almacén de frutos secos, de manera que quedan dispuestos los elementos `` y sus imágenes.

```

<main class="main">
  <section class="section" id="section1">
    <!-- 
  <section class="section" id="section2">
    <!-- 
  <section class="section" id="section3">
    <!-- 
  <section class="section" id="section4">
    <!-- 
  <section class="section" id="section5">
    <!-- 
</main>

```

- app/src/index.php

- Tras el tratamiento de los datos recibidos de ejemplo, rigurosamente iguales a los esperados y tras el estilo podemos ver el resultado de nuestra aplicación:
- Colaborativo:

← → ↻ pr200.newflow.tech 🔍 📄 ☆ 🌐 K (x) ⚙️ ☰ M ⋮

LABORAL AMAZON VENTA MATRICULAS „FOR... CURIOSITIES Daw Tools Referencias » Lista de lectura

Zonas de Almacenamiento

[Datos equipo >](#) [Datos prueba >](#)

Z100

Solicitud datos inaccesible.
Solicitud datos inaccesible.

Z200

Solicitud datos inaccesible.

2021-12-10 18:44:54temperatura21.7

2021-12-10 18:45:04 humedad 38.2

B201

2021-12-10 17:26:15 entradas 2

Z300

Solicitud datos inaccesible.
Solicitud datos inaccesible.

Z400

Solicitud datos inaccesible.

2021-12-10 10:05:56iluminacion41.87

B401

2021-12-10 18:45:44 entradas 15

B402

2021-12-10 18:45:44 entradas 8

Z500

Solicitud datos inaccesible.
Solicitud datos inaccesible.

API

- Datos propios

Zonas de Almacenamiento

[Datos equipo >](#)

[Datos prueba >](#)

Z100

2021-11-25 15:30 iluminacion 20
2021-11-25 15:30 temperatura 21
2021-11-25 15:30 humedad 50

A101

2021-11-25 15:30 pulsaciones 86

B101

2021-11-25 15:30 entradas 1

Z200

2021-11-25 15:30 iluminacion 90
2021-11-25 15:30 temperatura 26
2021-11-25 15:30 humedad 15

A201

2021-11-25 15:30 pulsaciones 8

A202

2021-11-25 15:30 pulsaciones 20

B201

2021-11-25 15:30 entradas 20

Elementos

Consola

1

Filtrar

Niveles predeterminados

1 problema: 1

Object script2.js:28

Z100 script2.js:29

Object script2.js:28

Z200 script2.js:29

Object script2.js:28

Z500 script2.js:29

Object script2.js:28

Z400 script2.js:29

Object script2.js:28

Z300 script2.js:29

> |

Consola

What's New

Problemas

Incluir problemas de cookies de terceros

0

0

1

Audit usage of navigator.userAgent, navigator.appVersion, and navigator.platform

Activar Windows

Ve a Configuración para activar Windows.

Zonas de Almacenamiento

[Datos equipo >](#) [Datos prueba >](#)



Gestión solicitudes fallidas

- Impresión de un mensaje de error, informando al cliente, que no se muestran en su totalidad las peticiones solicitadas.

```
.catch( function ( error ) {
    const pError = document.createElement("p");
    const smsError = document.createTextNode("Solicitud datos inaccesible.");
```

```

pError.appendChild(smsError);
pError.className = "smsError";
seccion.firstChild.appendChild( pError );
console.log(' Problema con la petición Fetch:' + error.message);
});

```



Promesas y uso de API interna urls

Estas API son para el uso propio de la aplicación no para terceros.

<https://pr200.newflow.tech/api/urlsPrueba.php>

<https://pr200.newflow.tech/api/urlsColab.php>

He Intentado de varias maneras, la gestión de una promesa que solicite JSON ofrecido por las APIs

y con el uso de la función de `pintasolicitudes()` `pr200\app\src\js\script1.js`

cada dos url recibidas pero no lo he conseguido, el ERROR que no he conseguido superar ha sido

Not Access firstElementChild

La intención era que al clicar en uno de estos dos link, se hiciera petición a API interna correspondiente y se solicitarán datos a las fuentes correspondientes .

Zonas de Almacenamiento

[Datos equipo >](#) [Datos prueba >](#)

```
linkdatequipo.addEventListener( 'click', usoUrl );
linkdatprueba.addEventListener( 'click', usoUrl );

function usoUrl( event ){
  // let pru = "https://pr200.newflow.tech/api/urlsPrueba.php"
  let pruebas = "https://newflow.tech/pr-200-jsonPruebas/allurl.php"
  let equipo = "https://pr200.newflow.tech/api/urlsColab.php"
  let request;
  console.log(event.currentTarget.id);
  if( event.currentTarget.id == "datprueba"){
    request = pruebas;
  }
  else{
    request = equipo;
  }
  request = pruebas;
```

```

console.log(request);
promiseUrlBase = fetch( request );
promiseUrlBase
.then( response => {
    return response.json();
})
.then( data => {
    console.log(data);
    const article = "article";
    let numarticle = 1;
    let accion = 0;
    data.forEach( (url, index) =>{
        if( index%2==0 ){
            val1 = url;
        }
        else{
            val2 = url;
            accion = 1;
        }
        if( accion==1 ){
            this[article+numarticle] = article+numarticle;
            console.log(val1); console.log(typeof val1);
            console.log(val2); console.log(typeof val2);
            console.log(this[article+numarticle]);
            console.log(typeof this[article+numarticle]);
            let articulo = document.getElementById("this[article+numarticle]");
            console.log(articulo);
            console.log(typeof articulo);
            pintasolicitudes( [val1, val2], articulo ); // ERROR EN ESTA LINEA !!!!
            accion = 0;
            numarticle++;
        }
    });
});
.catch( function ( error ) {
    console.log(' Problema con la petición Fetch total urls:' + error.message);
});
}

```

- Resuelvo poniendo directamente las urls.

```

pintasolicitudes ( ["http://arduino.proyectos-andres.tech/zonas/pr200/", "http://samuel080.me/api.php" ] , article1 );
pintasolicitudes ( ["http://pr200.rafadaw.software/app/mediciones/", "https://pr200.jose-velazquez-daw.rocks/api.php" ] , article2 );
pintasolicitudes ( ["http://manu080.me/js.php", "ballesteros....." ] , article3 );
pintasolicitudes ( ["pablo.....", "https://pr200.newflow.tech/api/api.php" ] , article4 );
pintasolicitudes ( ["http://ismaeldaw080.me/apiismael.php", "ocete....." ] , article5 );

```

Tests

Todos los tests de la aplicación se encuentran en app/tests

- Generando el archivo de conf

generar el archivo de configuracion de php unit

le indicamos a phphunit donde estan los tests y como correlos

```
$ ./bin/phpunit --generate-configuration
```

```
moi@EQ-600:~/proyectos/pr200$ docker exec -ti php-pr200 sh
/var/www/html # ls
LICENSE      bin          composer.lock src          vendor
README.md    composer.json enunciado.txt tests
/var/www/html # ./bin/phpunit --generate-configuration
PHPUnit 9.5.10 by Sebastian Bergmann and contributors.

Generating phpunit.xml in /var/www/html

Bootstrap script (relative to path shown above; default: vendor/autoload.php):
Tests directory (relative to path shown above; default: tests):
Source directory (relative to path shown above; default: src):
Cache directory (relative to path shown above; default: .phpunit.cache):

Generated phpunit.xml in /var/www/html.
Make sure to exclude the .phpunit.cache directory from version control.
```

- Ejecución de tests:

```
$ ./bin/phpunit --testdox
```

- Primer test fallado y superado.

```

Vehiculo (Moi\Tests\ApiTest\Vehiculo)
x Devuelve true si db instanciada es objeto [10.27 ms]
|
| bd no es objeto error!
| Failed asserting that mysqli Object &00000000000001680000000000000000 () is not of type "object".
|
| /var/www/html/tests/ApiTest.php:24
|

Time: 00:00.019, Memory: 6.00 MB

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
/var/www/html # ./bin/phpunit --testdox
PHPUnit 9.5.10 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.1.0RC5
Configuration: /var/www/html/phpunit.xml

Vehiculo (Moi\Tests\ApiTest\Vehiculo)
✓ Devuelve true si db instanciada es objeto [4.96 ms]

Time: 00:00.008, Memory: 6.00 MB

OK (1 test, 1 assertion)

```

- Tests para la clase Api realizados.

```

Vehiculo (Moi\Tests\ApiTest\Vehiculo)
✓ Devuelve true si class api retorna objeto [0.06 ms]
✓ Devuelve true si db instanciada es objeto [8.79 ms]
✓ Devuelve true si fx consultas retorna array [3.93 ms]
✓ Devuelve true si fx json retorna json esperado [5.85 ms]

Time: 00:00.022, Memory: 6.00 MB

OK (4 tests, 6 assertions)

```

- Test para nuevas clases class Bd y class ApiReceive


```
Runtime:      PHP 8.1.0RC5
Configuration: /var/www/html/phpunit.xml

Api Receive (Moi\Tests\ApiReceiveTest\ApiReceive)
  ✓ Devuelve true si class api receive retorna obj [1.42 ms]

Api (Moi\Tests\ApiTest\Api)
  ✓ Devuelve true si class api retorna objeto [2.44 ms]
  ✓ Devuelve true si fx consultas retorna array [3.60 ms]
  ✓ Devuelve true si fx json retorna json esperado [25.98 ms]

Bd (Moi\Tests\BdTest\Bd)
  ✓ Devuelve true si class bd test retorna objeto [1.45 ms]

Time: 00:00.038, Memory: 6.00 MB

OK (5 tests, 7 assertions)
```

Resumen usuario jca

EN RESUMEN:

Acceso servidor con subdominio <https://pr200.newflow.tech/>

user: jca

pass: jca

Acceso a base de datos del proyecto

user: jca

pass: jca

Usuario jca servidor y su llave pública

- Crear el usuario en el servidor **jca** con contraseña **daw** y que pertenezca al grupo **sudo**.

```
#remoto
$ sudo adduser jca
$ sudo addgroup jca sudo
```

```

:~ $ sudo adduser jca
Adding user `jca' ...
Adding new group `jca' (1001) ...
Adding new user `jca' (1001) with group `jca' ...
Creating home directory `/home/jca' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for jca
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
:~ $ sudo addgroup jca sudo
Adding user `jca' to group `sudo' ...
Adding user jca to group sudo
Done.
```

Cargar la clave **pública** "jca_daw_rsa.pub" en el `~/.ssh/authorized_keys` del usuario **jca**.

Desde servidor en AWSLab

```
#remoto
$ su - jca
$ cd ~
# crear directorio si no existe
$ mkdir .ssh
$ chmod 700 .ssh
$ cd .ssh
$ vim authorized_keys
#restringimos permisos authorized_keys
$ chmod go-rw authorized_keys
$ exit
```

```
jca@ip-172-31-29-185:~$ mkdir .ssh
jca@ip-172-31-29-185:~$ ls -la
total 24
drwxr-xr-x 3 jca jca 4096 Dec 7 14:17 .
drwxr-xr-x 5 root root 4096 Dec 7 14:10 ..
-rw-r--r-- 1 jca jca 220 Dec 7 14:10 .bash_logout
-rw-r--r-- 1 jca jca 3771 Dec 7 14:10 .bashrc
-rw-r--r-- 1 jca jca 807 Dec 7 14:10 .profile
drwxrwxr-x 2 jca jca 4096 Dec 7 14:17 .ssh
jca@ip-172-31-29-185:~$ chmod 700 .ssh
jca@ip-172-31-29-185:~$ cd .ssh/
jca@ip-172-31-29-185:~/ssh$ vim authorized_keys
jca@ip-172-31-29-185:~/ssh$ chmod go-rw authorized_keys
jca@ip-172-31-29-185:~/ssh$ ls -la
total 12
drwx----- 2 jca jca 4096 Dec 7 14:21 .
drwxr-xr-x 3 jca jca 4096 Dec 7 14:21 ..
-rw----- 1 jca jca 579 Dec 7 14:21 authorized_keys
jca@ip-172-31-29-185:~/ssh$ exit
logout
::~ $ whoami
ubuntu
```

Copiar el contenido de la llave publica de jca

```
jca@ip-172-31-29-185: ~/.ssh  
ssh -rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCm9H2G3y0h+KSOGQ/jPGYIMOL  
TjmOP2/Umio35uPnpLftEnXdG0AnWwy7qp/XGoLnt9S6ql0VxM3wlozzb1CxVFQ  
WPJX4s5Z/BwnYAs0i6qVrHapQAqFPqGMQLXop+VpNKV9CXmt63iqYCHU2JIzkr  
nn3wIH9ZsrBMoqk1XMepTtgYUDph+JKvt8buNCJsO+Can7j/FrvNFQUQu+VG3sa  
/7MOAkQ8GSq5yw3CrVMghYxQqfBbHpDwY8CNmrVJlfsAc4G+I8lePlzmU3LJGI  
eXdmCW4jtJyzUSdMBY398= robotica@LAPTOP-B91ABPH
```

~
~
~
~
~
~
~
~
wa!

Usuario jca mysql y permisos base de datos

Creo el usuario **jca** con acceso a mysql y permisos para poder trabajar en la **base de datos** donde está el proyecto.

- Como no consigo crear los usuarios gestores de la base de datos, automaticamente desde docker en fichero Dockerfile, los realizaremos manualmente una vez levantado el proyecto.
- Para esto entraremos al contenedor de mysql

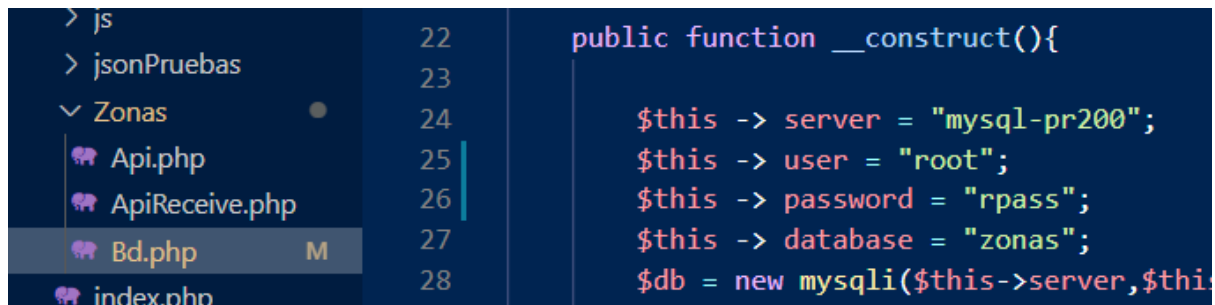
```
#remoto
$ docker exec -ti mysql-pr200 sh
$ mysql -u root -p
$ rpass

$ CREATE USER `moidb` IDENTIFIED BY 'moidbpass';
$ GRANT ALL ON zonas.* TO 'moidb'@'%';
$ FLUSH PRIVILEGES;

$ CREATE USER `jca` IDENTIFIED BY 'jca';
$ GRANT ALL ON zonas.* TO 'jca'@'%';
$ FLUSH PRIVILEGES;

$ exit
```

- Tras estos pasos podremos cambiar el usuario que gestiona la base de datos zonas, por cualquiera de estos usuarios creados para la gestión específica de esta base de datos, por defecto el usuario que accederá a la base de datos es root, NO olvides modificarlo en producción.



pr200/app/src/Zonas/Bd.php