

# Aplicación para kiosko informativo usando tarjetas NFC de estudiantes



Javier Mariscal, Rafael Castellero y Miguel Angel Pintado

# Índice

- Estado y Objetivos
- Esquema de red y Recursos Hardware/Software
- Implementación en DOCKER
- Diseño Gráfico
- Diseño BBDD
- Estructura y Programación
- Programación API Shikoba
- Conexión API y Comprobaciones
- Objetivos Cumplidos/No cumplidos
- Conclusiones
- Bibliografías

# ESTADO Y OBJETIVOS

The background is a dark blue gradient. It features several abstract geometric elements: a large triangle in the top left, a network of lines and dots in the top center, a triangle in the top right, a small triangle in the middle right, a triangle in the bottom left, a series of nested triangles in the bottom center, and a large triangle in the bottom right.

# Estado y Objetivos

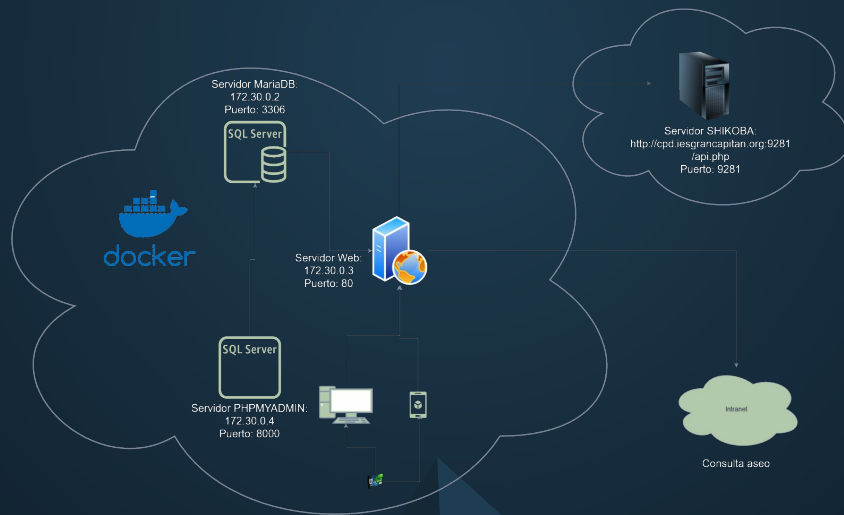
## ESTADO

- Partimos de una web ya creada pero mal optimizada
- Sin un entorno en el que estructurar la web
- Sin apartado de convivencia

## OBJETIVOS

- Optimización del código
- Creación de docker para almacenar los servicios
- Conexión con intranet para salida al aseo
- Conexión con Shikoba para comprobar puntos de alumnado
- Comprobar en Shikoba las sanciones
- Conexión con APIs
- Mostrar Información
- Indicar si un profesor es administrador a través de la web
- Queremos evitar estas pérdidas de tiempo.
- Permitir que se pueda usar desde cualquier parte del centro educativo
- Se pueda usar en cualquier dispositivo que disponga el profesorado

# Esquema de Red y Recursos Hardware/Software



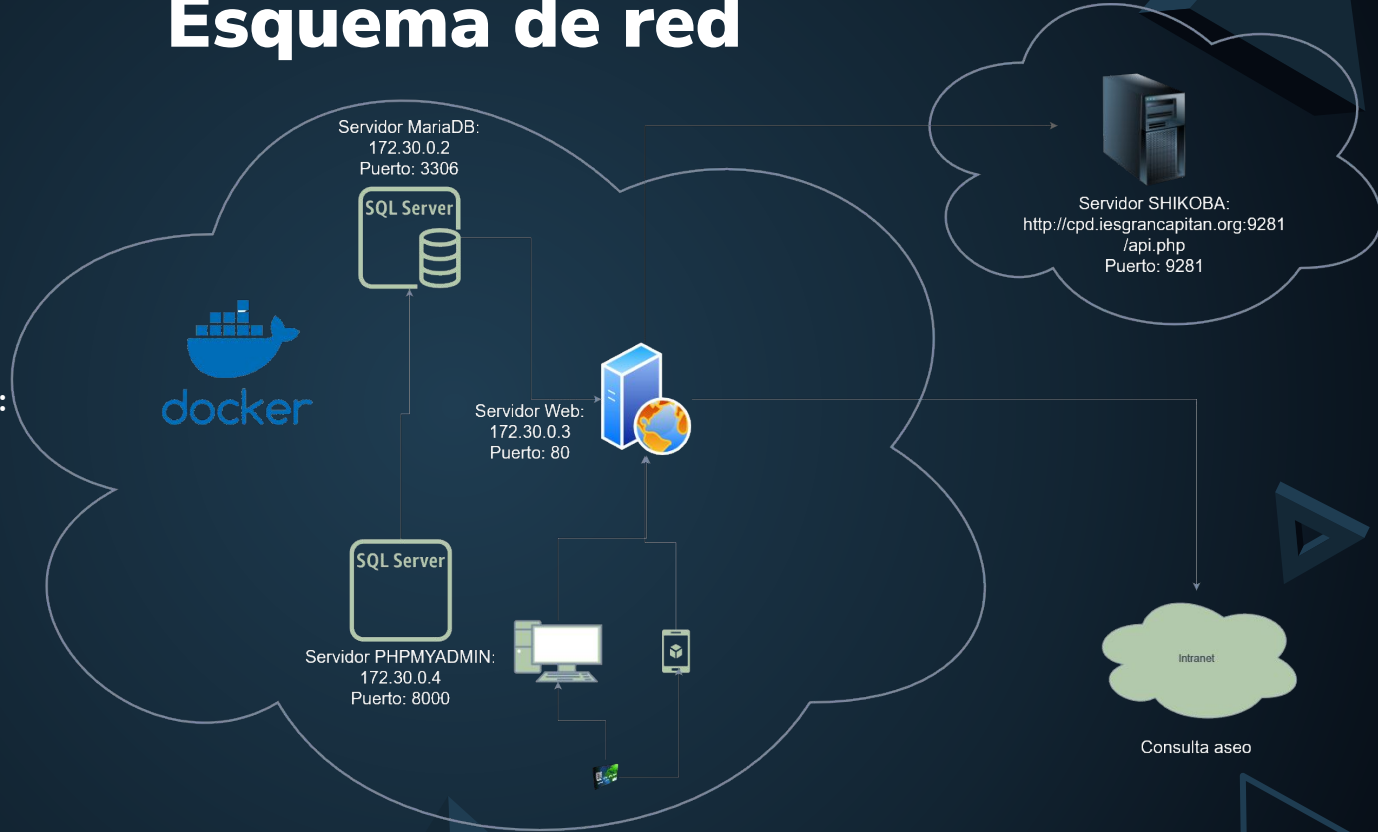
# Esquema de red

## Servidores:

- Web
- Mysql
- PHPMYADMIN
- SHIKOBA

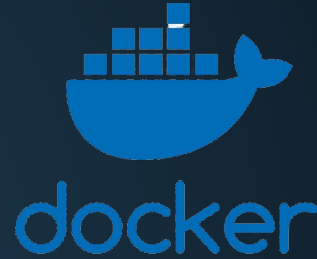
## Puertos de los Servicios:

- 3306
- 80
- 8000
- 9281



# RECURSOS HARDWARE

- Versión de docker: 4.12.0
- Lector NFC
- Tarjetas NFC



# RECURSOS SOFTWARE



- Sistemas Operativos: Debian 10
- PHP: Versiones 7.4 y 8.0
- Apache: Versiones 2.4.54
- MARIADB: Versiones 10.4.27
- Bases de datos en MARIADB
- Programas de desarrollo: VISUAL STUDIO CODE.





# DOCKER

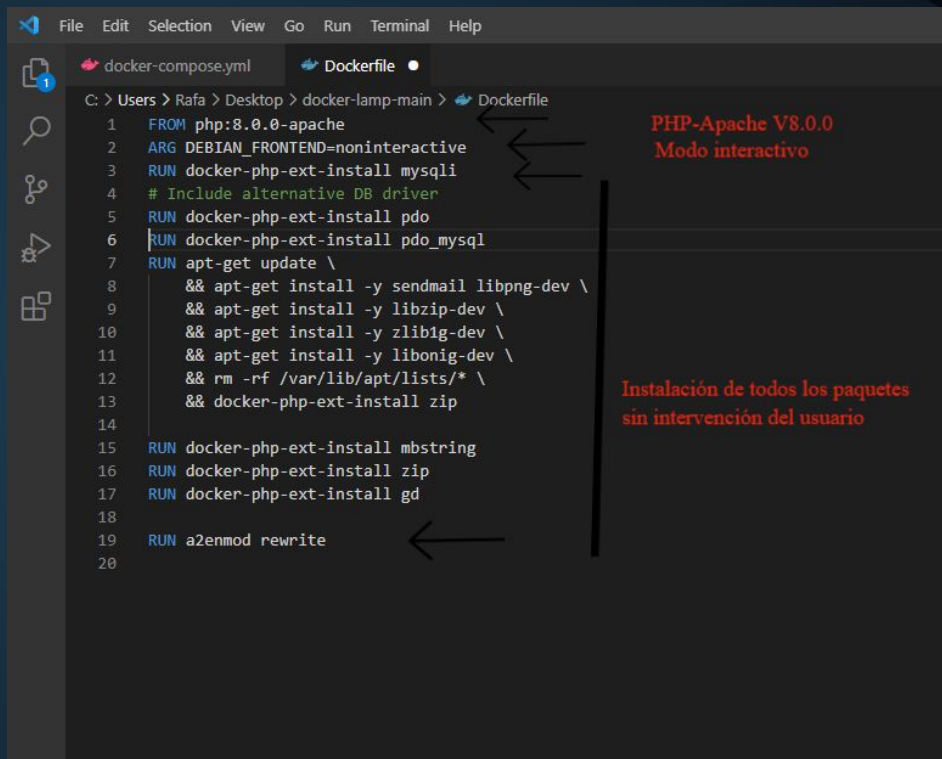


# DOCKERFILE

Servicios:

- Apache
- PHP

Instalación de paquetes de forma no interactiva.



```
File Edit Selection View Go Run Terminal Help
docker-compose.yml Dockerfile
C: > Users > Rafa > Desktop > docker-lamp-main > Dockerfile
1 FROM php:8.0.0-apache
2 ARG DEBIAN_FRONTEND=noninteractive
3 RUN docker-php-ext-install mysqli
4 # Include alternative DB driver
5 RUN docker-php-ext-install pdo
6 RUN docker-php-ext-install pdo_mysql
7 RUN apt-get update \
8     && apt-get install -y sendmail libpng-dev \
9     && apt-get install -y libzip-dev \
10    && apt-get install -y zlib1g-dev \
11    && apt-get install -y libonig-dev \
12    && rm -rf /var/lib/apt/lists/* \
13    && docker-php-ext-install zip
14
15 RUN docker-php-ext-install mbstring
16 RUN docker-php-ext-install zip
17 RUN docker-php-ext-install gd
18
19 RUN a2enmod rewrite
20
```

PHP-Apache V8.0.0  
Modo interactivo

Instalación de todos los paquetes  
sin intervención del usuario

# Docker Compose

- Servicios: DB, WWW, PHPMYADMIN.
- Image: Ruta de donde se descarga el contenedor.
- Puertos: Puertos de conexión.
- Command: Comando por defecto.
- Enviroment: Variables de entorno.
- Volúmenes: Copia de seguridad.
- Networks: Nombre y IP.

```
docker-compose.yml X
C > Users > Rafa > Desktop > Proyecto > docker-lamp-main > docker-compose.yml

1 version: "3.1"
2 services:
3   db:
4     image: rafacasti/proyectotarjetanfc:mysqlV1
5     ports:
6       - "3306:3306"
7     command: --default-authentication-plugin=mysql_native_password
8     environment:
9       MYSQL_DATABASE: grancapitan
10      MYSQL_PASSWORD: test
11      MYSQL_ROOT_PASSWORD: test
12     volumes:
13       - ./dump:/docker-entrypoint-initdb.d
14       - ./conf:/etc/mysql/conf.d
15       - persistent:/var/lib/mysql
16     networks:
17       tarjetaNFC:
18         ipv4_address: 172.30.0.2
19
20   www:
21     image: rafacasti/proyectotarjetanfc:wwwV2
22     ports:
23       - "80:80"
24     links:
25       - db
26     networks:
27       tarjetaNFC:
28         ipv4_address: 172.30.0.3
29
30   phpmysqladmin:
31     image: rafacasti/proyectotarjetanfc:phpmyadminV1
32     links:
33       - db:db
34     ports:
35       - 8000:80
36     environment:
37       MYSQL_USER: root
38       MYSQL_PASSWORD: test
39       MYSQL_ROOT_PASSWORD: test
40     networks:
41       tarjetaNFC:
42         ipv4_address: 172.30.0.4
43
44   volumes:
45     persistent:
46
47   networks:
48     tarjetaNFC:
49       ipam:
50         config:
51           - subnet: "172.30.0.0/16"
52           gateway: "172.30.0.1"
```

Declaramos los servicios  
Servicio  
Imagen MySQL  
Puerto de salida  
Comando por defecto  
Variables de entorno  
Copia de seguridad  
Nombre de la red  
IP del Contenedor  
Imagen Apache  
Puerto de salida  
Enlace de conexión  
Nombre de la red  
IP del Contenedor  
Imagen PHPMYADMIN  
Enlace de conexión  
Puerto de salida  
Variables de entorno  
Nombre de la red  
IP del Contenedor  
Red  
Puerta de enlace

# DISEÑO GRÀFICO

The background is a dark blue gradient. It features several abstract geometric elements: a large 3D triangle in the top left, a network of lines and dots in the top center, a 3D triangle in the top right, a 2D triangle in the middle right, a 3D triangle in the bottom left, a 3D triangle in the bottom center, and a 2D triangle in the bottom right.

# Interfaz gráfica

## Selección de tarjeta

ADMINISTRACIÓN

Profesor Alumno Inicio

ADMINISTRACIÓN

Elige una unidad

2ºGSDAWA ▾

Mostrar Volver Inicio

ADMINISTRACIÓN

LISTADO DE USUARIOS DE 2ºGSDAWA

Correo/Id	Tarjeta
<input type="radio"/> 3354132	Tarjeta no asignada
<input checked="" type="radio"/> 2293874	Tarjeta no asignada
<input type="radio"/> 752252	80543219
<input type="radio"/> 8392050	65631235
<input type="radio"/> 3093004	77725507

Mostrar Volver Inicio

ADMINISTRACIÓN

Usuario

2293874

Asigna una tarjeta

Asignar tarjeta Inicio

# Interfaz gráfica

## Borrado de tarjeta

### BORRADO DE TARJETA ASOCIADA

Elige un departamento o una unidad

2°GSDALWA

Mostrar Inicio

### BORRADO DE TARJETA ASOCIADA

#### LISTADO DE USUARIOS

Correo/Id	Tarjeta
3354332	Tarjeta no asignada
2293074	Tarjeta no asignada
<input type="checkbox"/> 752252	80543210
<input type="checkbox"/> 8392050	65631235
<input type="checkbox"/> 3003004	77725507

Borrar Nueva consulta Inicio

### MOTIVO DE LA BAJA

Baja matricula

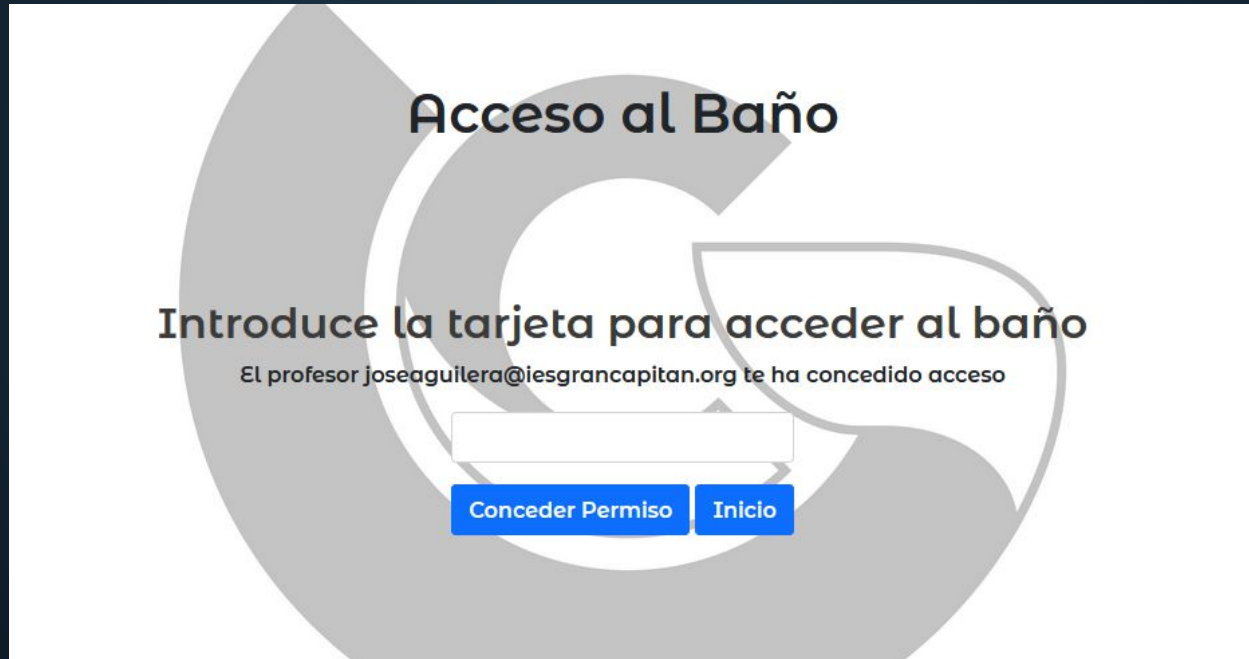
Borrar

Nueva consulta

Inicio

# Interfaz gráfica

Borrado de tarjeta



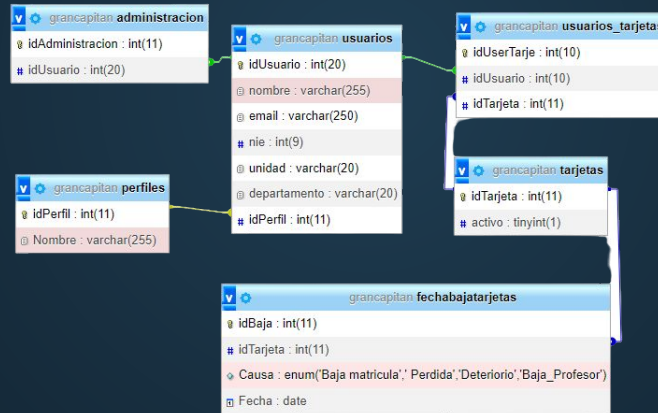
**Acceso al Baño**

**Introduce la tarjeta para acceder al baño**

El profesor joseaguilera@iesgrancapitan.org te ha concedido acceso

[Conceder Permiso](#) [Inicio](#)

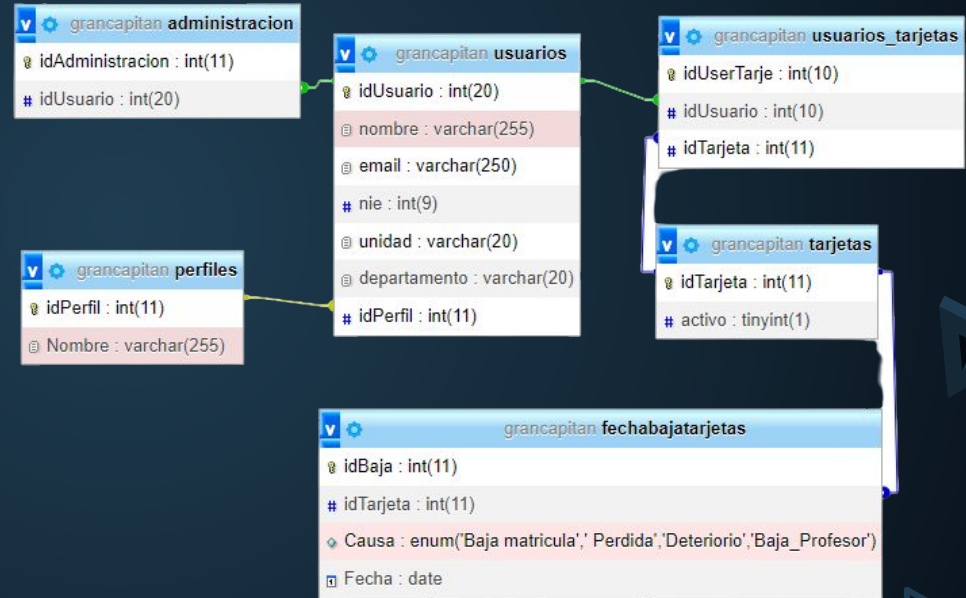
# DISEÑO BBDD





# BBDD

- Se han realizado cambios en la anterior BBDD para actualizarla.
- Profesores y alumnos => Usuarios
- Se ha añadido la tabla perfiles y administrador.



The background is a dark blue gradient. It features several abstract geometric elements: a network of thin white lines with small circular nodes at the top center; a large, dark blue 3D triangle in the top right corner; a smaller, lighter blue 3D triangle in the top left corner; a small, light blue 3D triangle in the bottom left corner; a small, light blue 3D triangle in the bottom right corner; and a series of concentric, slightly offset triangles at the bottom center, creating a tunnel-like effect.

# ESTRUCTURA Y PROGRAMACIÓN

# ESTRUCTURA MVC

## ¿Qué es MVC?

MVC es el Modelo Vista Controlador, es un patrón que se usa para tener cierto orden y escalabilidad en los desarrollos

Está compuesto por:

- **Modelo:** se encarga de, por ejemplo, cargar datos y realizar operaciones en ellos.
- **Vista:** el **front end o interfaz gráfica** de usuario (GUI), es la capa que vé el usuario.
- **Controlador:** se encarga de solicitar datos al modelo y enviarlos a la vista. Interconecta el frontend con el backend.

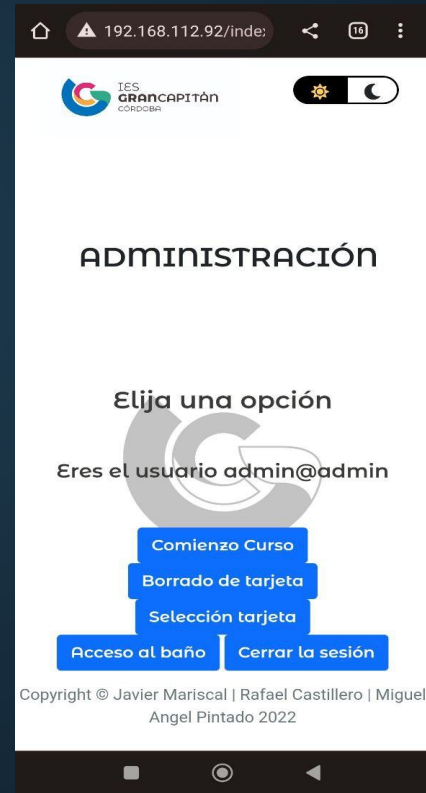
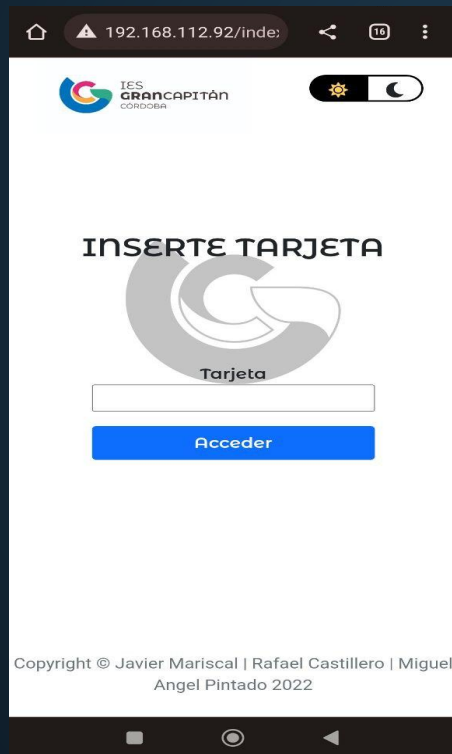
# Programación de la API

Los lenguajes utilizados han sido

- PHP
- HTML
- CSS
- JAVASCRIPT



# WEB RESPONSIVE



# Programación de la WEB

Al usar el MVC se ha tenido que modificar algunos archivos de la API, entre ellos el index.

```
<?php

session_start();
require_once 'config/config.php';
require_once 'model/conexionDB.php';

if (!isset($_GET["controller"])) $_GET["controller"] = constant("DEFAULT_CONTROLLER");
if (!isset($_GET["action"])) $_GET["action"] = constant("DEFAULT_ACTION");

$controller_path = 'controller/' . $_GET["controller"] . '.php';

/* Check if controller exists */
if (!file_exists($controller_path)) $controller_path = 'controller/' . constant("DEFAULT_CONTROLLER") . '.php';

/* Load controller */
require_once $controller_path;
$controllerName = $_GET["controller"] . 'Controller';
$controller = new $controllerName();

/* Check if method is defined */
$dataToView["data"] = array();
if (method_exists($controller, $_GET["action"])) $dataToView["data"] = $controller->{_GET["action"]}();

/* Load views */
require_once 'view/template/header.php';
require_once 'view/' . $controller->view . '.php';
require_once 'view/template/footer.php';
```

En este fichero recibiremos todas las peticiones, tanto para el controlador de usuario como para otros si los hubiera..

# Programación de la WEB

Como hemos indicado antes, en el *model* realizaremos las consultas a la BBDD, al cual crearemos las clases necesarias y las funciones pertinentes para su ejecución

```
9 references | 0 overrides
public function getConnection()
{
    $dbObj = new conexiondb();
    $this->connection = $dbObj->initConex();
}

/* Save note */
1 reference | 0 overrides
public function isAdmin(int $idUser): bool
{
    $this->getConnection();

    $query = $this->connection->query("SELECT * FROM administracion WHERE idUsuario=" . $idUser . ";");

    return $query->num_rows === 1;
}
```



# Programación de la WEB

El controller se encarga de recibir las peticiones desde la vista, solicitar información y/u ordenar cambios al modelo.

```
require_once 'model/tarjeta.php';
require_once 'model/usuario.php';

class UsuarioController
{
    public $page_title;
    public $page_error;
    public $page_success;
    public $view;

    public function __construct()
    {
        if (!isset($_SESSION['user'])) {
            header('Location: index.php');
        }
        $this->view = 'user_list';
        $this->tarjetaObj = new Tarjeta();
        $this->usuarioObj = new Usuario();
    }
}
```

```
public function listado()
{
    $this->select = $_POST['select'] ?? null;
    $this->listado = [];

    if (null !== $this->select) {
        $type = explode('_', $this->select)[0];
        $value = explode('_', $this->select)[1];
        $this->listado = $this->usuarioObj->getFilterUser($type, $value);
    }

    $this->departamentos = $this->usuarioObj->getDepartamentos();
    $this->unidades = $this->usuarioObj->getUnidades();
    $this->page_title = 'BORRADO DE TARJETA ASOCIADA';
}
```



# Programación de la WEB

El view se encargará de mostrar los datos recibidos del controlador.

```
<div id='divcentral' class='container px-4 px-lg-5 d-flex h-100 align-items-center justify-content-center'>
  <div class='d-flex justify-content-center'>
    <div id='divcentral2' class='text-center'>
      <form method='post' action=index.php?controller=tarjeta&action=borradoAsociacion>
        <div class='container'>
          <select name="motivo" class="form-select">
            <option name='bajaMatricula' value="Baja matricula">Baja matricula</option>
            <option name='Perdida' value="Perdida" >Perdida</option>
            <option name='Perdida' value="Deteriorio">Deteriorio</option>
            <option name='Baja_Profesor' value="Baja_Profesor">Baja Profesor</option>
          </select>
          <?php foreach ($controller->usuarios as $idUserario) { ?>
            <input type="hidden" name="usuarios[]" value="<?php echo $idUserario ?>">
          <?php } ?>
        </div>
        <div class="mt-3">
          <input type='submit' class='btn btn-primary' name='borrarAsociacion' value='Borrar'>
          <a class='btn btn-primary' href='index.php?controller=usuario&action=listado'>Nueva consulta</a>
          <a class='btn btn-primary' href='index.php'>Inicio</a>
        </div>
      </form>
    </div>
  </div>
</div>
```

# Programación de la WEB

Para la salida al aseo se ha modificado para que sea acorde a la nueva estructura.

```
public function permiso_bano()  
{  
    $this->profesor = $_SESSION['user']['email'];  
    $tarjeta = $_POST['idTarjeta'];  
    if (!$tarjeta) {  
        $this->view = 'acceso_bano';  
        $this->page_title = 'Acceso al Baño';  
        $this->page_error = 'Introduce una tarjeta correcta';  
        return;  
    }  
    $nieAlumno = $this->tarjetaObj->comprobarUserBano($tarjeta);  
  
    if (!$nieAlumno) {  
        $this->view = 'acceso_bano';  
        $this->page_title = 'Acceso al Baño';  
        $this->page_error = 'El usuario no existe';  
        return;  
    }  
}
```

```
$url = "http://cpd.iesgrancapitan.org:9280/api.php?nie=  
{ $nieAlumno['nie'] }&profe=$this->profesor";  
$curl = curl_init();  
curl_setopt($curl, CURLOPT_URL, $url);  
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);  
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type:application/json'));  
curl_setopt($curl, CURLOPT_HTTPGET, true);  
$data = curl_exec($curl);  
if (trim($data) == "SI") {  
    $this->view = 'aseo_libre';  
    $this->page_title = 'Acceso Concedido';  
} else {  
    $this->view = 'aseo_ocupado';  
    $this->page_title = 'Acceso Denegado';  
}
```

# PROGRAMACIÓN API-SHIKOBÁ

# Programación API-SHIKOB

```
<?php
//echo "hola mundo";
include("../conexion.php");
/**
 * Metodo Ruta
 * GET /permiso Obtener permiso o no para ir al aseo
 */
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: OPTIONS,GET,POST");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");
#Variable para respuesta.
$response['status_code_header']='HTTP/1.1 404 Not Found';
$response['body']=null;
#Metodo usado
$requestMethod = $_SERVER['REQUEST_METHOD'];
#Parsear direccion de entrada
$request =parse_url( $_SERVER['REQUEST_URI'], PHP_URL_PATH);
#var_dump($request);
#uri = explode('/', $request);
#isset($uri[2])) { }
$nie=$_GET['nie'];
$result= "NO";
$error= "";
switch ($requestMethod){
case 'GET':
    // CONSULTA1: sacar el profesor nombre a partir de su user
    $sqlpuntos = "SELECT puntos, nombre, apellido1, apellido2 from alumno where nie = " . $nie ;
    //$result = $sqlpuntos;
    $result=mysqli_query($conn, $sqlpuntos);
    $data=mysqli_fetch_array($result);
    if( mysqli_num_rows($result) > 0){
        $row = mysqli_fetch_assoc($result);
        $nombre=$row['nombre'];
        $puntos=$row['puntos'];
        $apellido1=$row['apellido1'];
        $apellido2=$row['apellido2'];
    }
    else {
        $result="no hay datos";
    }
break;
default:
;
}
// CONTROLAR ERRORES DE BBDD
$result=$data;
$response['body']=json_encode($result);
if ($error=="") {
    $response['status_code_header']='HTTP/1.1 200 OK';
}
#Escribir la salida
header($response['status_code_header']);
if ($response['body']){
    echo $response['body'];
}
```

Obtención de  
puntos

# Programación API-SHIKOB

```
<?php
//echo "hola mundo";
include("../conexion.php");
/**
 * Metodo Ruta
 * GET /permiso Obtener permiso o no para ir al aseo
 */
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: OPTIONS,GET,POST");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");
#Variable para respuesta.
$response['status_code_header']='HTTP/1.1 404 Not Found';
$response['body']=null;
#Metodo usado
$requestMethod = $_SERVER['REQUEST_METHOD'];
#Parsear direccion de entrada
$request = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
#var_dump($request);
#uri = explode('/', $request);
#isset($uri[2]) { }
$nie=$_GET['nie'];
$result="NO";
$error="";
switch ($requestMethod){
case 'GET':
    // CONSULTA1: sacar el profesor_nombre a partir de su user
    $sql= "SELECT fecha, fechaInicio, fechaFinal, sancion, tipo_sancion.tipo FROM sanciones inner join alumno on sanciones.idAlumno = alumno.id INNER JOIN tipo_sancion
    // $sql = "SELECT fecha, fechaInicio, fechaFinal, sancion FROM sanciones inner join alumno on sanciones.idAlumno = alumno.id WHERE sanciones.idEstado='4' and $
    $result=mysqli_query($conn, $sql);
    $data=mysqli_fetch_all($result);
break;
default:
;
}
// CONTROLAR ERRORES DE BBDD
$result=$data;
$response['body']=json_encode($result);
if ($error=="") {
    $response['status_code_header']='HTTP/1.1 200 OK';
}
#Escribir la salida
header($response['status_code_header']);
if ($response['body']){
    echo $response['body'];
}
```

## Sanciones actuales

# Programación API-SHIKOB

```
<?php
include("../conexion.php");
/**
 * Metodo Ruta
 * GET /permiso Obtener permiso o no para ir al aseo
 */
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Methods: OPTIONS,GET,POST");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");
#Variable para respuesta.
$response['status_code_header']='HTTP/1.1 404 Not Found';
$response['body']=null;
#Metodo usado
$requestMethod = $_SERVER['REQUEST_METHOD'];
#Parsear direccion de entrada
$request = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
#var_dump($request);
#$uri = explode('/', $request);
# if(isset($uri[2])) { }
$nie=$_GET['nie'];
$result= "NO";
$error= "";
switch ($requestMethod){
case 'GET':
    // CONSULTA1: sacar el profesor_nombre a partir de su user
    $sql = "SELECT fecha, fechaInicio, fechaFinal, sancion, tipo_sancion.tipo FROM sanciones inner join alumno on sanciones.idAlumno = alumno.id INNER JOIN tipo_sancion";
    // $result = $sqlpuntos;
    $result=mysqli_query($conn, $sql);
    $data=mysqli_fetch_all($result);
break;
default:
;
}
// CONTROLAR ERRORES DE BBDD
$result=$data;
$response['body']=json_encode($result);
if ($error=="") {
    $response['status_code_header']='HTTP/1.1 200 OK';
}
#Escribir la salida
header($response['status_code_header']);
if ($response['body']){
    echo $response['body'];
}
}>
```

## Histórico de sanciones



# **Conexión a la API y comprobaciones**

# Conexión a Shikoba

En la parte del código hace una consulta a la api y muestra los puntos .

```
<?php
include('alumnnav.php');
// Creamos la consult
$url = "http://192.168.0.141:8080/api.php?nie=". $_SESSION['user']['nie'];
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, $url );
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type:application/json'));
curl_setopt($curl, CURLOPT_HTTPGET, true);
$data = curl_exec($curl);
$data=json_decode($data, true);
//echo '<pre>',print_r($data,1),'</pre>';
$nombre=$data['nombre'];
$apellidos= $data[2].' '. $data[3];
$puntos=$data[0];
curl_close($curl);
?>
<html>
<body>
<div>
<h1 class="n" style="margin-top: -70px !important;"><?php echo $nombre . ' ' . $apellidos ?></h1>
<div class="dot">
<h1 class="puntos"><?php echo $puntos ?>/10</h1>
</div>
</div>
</body>
```





# Conexión a Shikoba

En la parte del código hace una consulta a la api y la decodifica en modo de array y la muestra en la página web.

```
<html>
<?php
include('alumnnav.php');
// Creamos la consulta
$url = "http://192.168.0.141:8080/sancionactual.php?nie=". $_SESSION['user']['nie'];
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, $url );
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type:application/json'));
curl_setopt($curl, CURLOPT_HTTPGET, true);
$data = curl_exec($curl);
$data=json_decode($data, true);
?>
<body>
<h1>SANCIONES ACTIVAS</h1>
<div class="container">
<div class="row border border-dark">
<div class="col-sm">Descripción</div>
<div class="col-sm">Tipo</div>
<div class="col-sm">Fecha</div>
<div class="col-sm">Fecha Inicio</div>
<div class="col-sm">Fecha Final</div>
</div>
<div>
<?php
foreach($data as $value){
    $fecha = $value[0];
    $fechainicio = $value[1];
    $fechafin = $value[2];
    $descripcion = $value[3];
    $tipo = $value[4];
    echo "<div class='row border-bottom border-dark'>";
    echo "<div class='col-sm'>" . $descripcion . "</div>";
    echo "<div class='col-sm'>" . $tipo . "</div>";
    echo "<div class='col-sm'>" . $fecha . "</div>";
    echo "<div class='col-sm'>" . $fechainicio . "</div>";
    echo "<div class='col-sm'>" . $fechafin . "</div>";
    echo "</div>";
}
curl_close($curl);
?>
</div>
</body>
<style>
```



[Inicio](#) [Sancion actual](#) [HISTORICO DE SANCIONES](#) [Cerrar sesion](#)

## SANCIONES ACTIVAS

Descripción	Tipo	Fecha	Fecha Inicio	Fecha Final
Expulsión aula de convivencia	Horas AC	2022-09-30 13:27:00	2022-09-30 14:27:00	2022-09-30 14:27:00

# Conexión a Shikoba

Para recoger varios datos en forma de array utilizamos un foreach para que recorra el array

```
<html>
<?php
include('alumnav.php');
// Creamos la consulta
$url = "http://192.168.0.141:8080/sanciones.php?nie=". $_SESSION['user']['nie'];
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type:application/json'));
curl_setopt($curl, CURLOPT_HTTPGET, true);
$data = curl_exec($curl);
$data=json_decode($data, true);
?>
<body>
<h1>LISTADO DE SANCIONES</h1>
<div class='container'>
<div class='row border border-dark'>
<div class='col-sm'>Descripcion</div>
<div class='col-sm'>Tipo</div>
<div class='col-sm'>Fecha</div>
<div class='col-sm'>Fecha Inicio</div>
<div class='col-sm'>Fecha Final</div>
</div>
<div>
<?php
foreach($data as $value){
    $fecha = $value[0];
    $fechainicio = $value[1];
    $fechafin = $value[2];
    $descripcion = $value[3];
    $tipo = $value[4];
    echo "<div class='row border-bottom border-dark'>";
    echo "<div class='col-sm'> . $descripcion .</div>";
    echo "<div class='col-sm'> . $tipo .</div>";
    echo "<div class='col-sm'> . $fecha .</div>";
    echo "<div class='col-sm'> . $fechainicio .</div>";
    echo "<div class='col-sm'> . $fechafin .</div>";
    echo "</div>";
}
curl_close($curl);
?>
</div>
</body>
```



Descripcion	Tipo	Fecha	Fecha Inicio	Fecha Final
Expulsión aula de convivencia	Horas AC	2022-09-30 13:27:00	2022-09-30 13:27:00	2022-09-30 14:27:00
Expulsión aula de convivencia	Horas AC	2022-09-30 14:17:00	2022-09-30 14:17:00	2022-09-30 15:17:00

# ¿Qué se quiere solucionar con este proyecto?

Hemos observado que el sistema actual es más que deficiente ya que obliga a los profesores a estar pendientes de las salidas del aula y de los alumnos que salen se encuentran el aseo ocupado.

- Queremos evitar estas pérdidas de tiempo.
- Permitir que se pueda usar desde cualquier parte del centro educativo
- Se pueda usar en cualquier dispositivo que disponga el profesorado

# Objetivos

## Cumplidos

- Optimización del código.
- Montaje de entorno en docker.
- Obtener puntuación de alumnado en shikoba
- Obtener un histórico de sanciones del curso.

## No cumplidos

- Gestión de administradores de la app: la tabla "**Administración**" en la que se indicará que *usuarios-profesores* serán administradores. Actualmente se indica directamente en la tabla qué usuario es administrador.
- Subida al servidor Proxmox.
- Opción de "Información general del centro" al kiosko informativo.

# Conclusiones

Problemas encontrados:

- Falta de Estructura.
- Código Antiguo.
- Falta de compatibilidad entre el KIOSKO y SHIKOBA.
- Problemas con la web en DOCKER.
- Problemas con los datos de la base de datos.

# Conclusiones

## FUTURAS MEJORAS:

- Poder usarlo para la entrada/salida del instituto en los recreos.
- Que indique el tiempo que queda para que esté libre el aseo.
- Añadir usuarios a través de la web individualmente.
- Aparte de mostrar las sanciones que se puedan añadir manualmente.

# BIBLIOGRAFÍAS

- ADAWEB <https://www.adaweb.es/modelo-vista-controlador-mvc-en-php-actualizado-2022/>
- PHP.NET <https://www.php.net/>
- STACKOVERFLOW <https://stackoverflow.com/>
- SOMOSPNT <https://somospnt.com/blog/252-como-puedo-crear-un-volumen-en-docker>
- DOCKERTIPS <https://dockertips.com/utilizando-docker-compose>
- COLABORATORIO  
<https://colaboratorio.net/davidochobits/sysadmin/2018/crear-imagenes-medida-docker-dockerfile/>

The background is a dark blue gradient. It features several abstract geometric elements: a network of thin white lines with small circular nodes at the top center; a large, dark blue 3D triangle in the top right corner; a smaller, dark blue 3D triangle in the top left corner; a small, dark blue 3D triangle in the bottom left corner; a small, dark blue 3D triangle in the bottom center; and a small, dark blue 3D triangle in the bottom right corner. The text "GRACIAS POR SU ATENCIÓN" is centered in a large, white, sans-serif font.

GRACIAS POR SU  
ATENCIÓN