

Tarjetas NFC

Javier Mariscal, Rafael Castellero y Miguel Angel
Pintado

Índice

- Instalación de los contenedores
- Estructura MVC
- Modificaciones en la BBDD
- Programación de la API
- Programación de la API Shikoba
- ¿Qué se quiere solucionar con este proyecto?
- Recursos
- Conclusiones
- Bibliografías

Instalación de los contenedores

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
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   www:
20     image: rafacasti/proyectotarjetanfc:wwwV2
21     ports:
22       - "80:80"
23     links:
24       - db
25     networks:
26       tarjetaNFC:
27         ipv4_address: 172.30.0.3
28   phpmysqladmin:
29     image: rafacasti/proyectotarjetanfc:phpmyadminV1
30     links:
31       - db:db
32     ports:
33       - 8000:80
34     environment:
35       MYSQL_USER: root
36       MYSQL_PASSWORD: test
37       MYSQL_ROOT_PASSWORD: test
38     networks:
39       tarjetaNFC:
40         ipv4_address: 172.30.0.4
41   volumes:
42     persistent:
43   networks:
44     tarjetaNFC:
45     ipam:
46       config:
47         - subnet: "172.30.0.0/16"
48         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

SHIKOBA

Nombre del servicio: Api shikoba.

Imagen: Ruta donde la descarga.

Nombre del contenedor: apishikoba.

Link: Vinculado con el contenedor de la WEB.

Puertos: Puertos de conexión.

Volúmenes: Donde se van a compartir los ficheros desde el anfitrión con el contenedor.

Network: Nombre de la red y IP.

```
api_shikoba:
  image: rafacasti/proyectotarjetanfc:apishikobaV2
  container_name: apishikoba
  links:
    - www
  ports:
    - "3307:3306"
    - "81:80"
  volumes:
    - ./dump_shikoba:/docker-entrypoint-initdb.d
    - ./conf_shikoba:/etc/mysql/conf.d
    - ./html_shikoba:/var/www/api-shikoba
  networks:
    tarjetaNFC:
      ipv4_address: 172.30.0.5
```

Servicio

Nombre del contenedor

Vinculación con otro contendor

Puertos

Volumenes

Ip de la maquina

ESTRUCTURA MVC

¿Que es MVC?

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

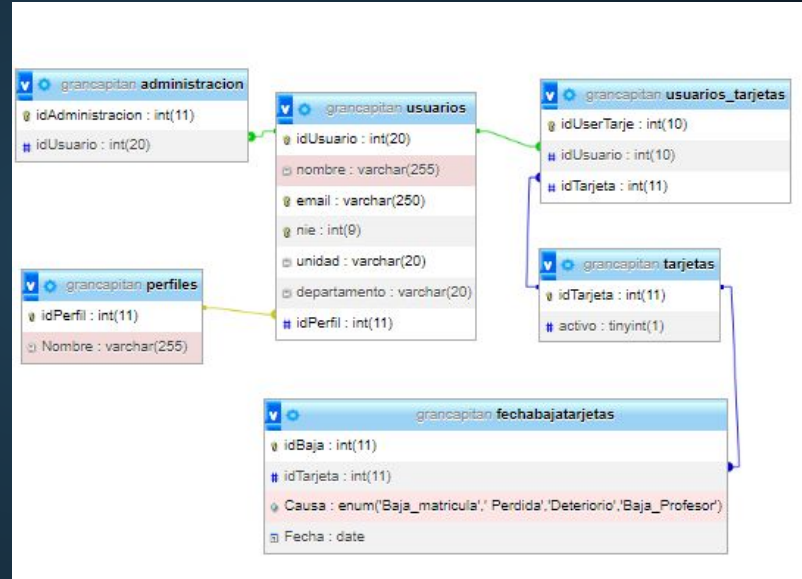
Está compuesto por:

- **Modelo:** se encarga de, por ejemplo, cargar datos y realizar operaciones en ellos.
- **Vista:** el **frontend 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.

MODIFICACIONES EN LA BBDD

Se han realizado varias modificaciones con respecto a la BBDD anterior

- Se ha cambiado la tabla alumnos y profesores por USUARIOS
- Se ha añadido la tabla perfiles
- se ha añadido la tabla administración



Programación de la API

Los lenguajes utilizados han sido

- PHP
- HTML
- CSS
- JAVASCRIPT



Programación de la API

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 API

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 $idUserario): bool
{
    $this->getConnection();

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

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

Programación de la API

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 API

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 API

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 de la API 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.

```
v.php');  
// Creamos la consult  
$url = "http://192.168.1.96:8080/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);
```



IES GRANCAPITAN CORDOBA

Inicio Sancion actual HISTORICO DE SANCIONES Cerrar sesion

Luna Molina Crespo

10/10

copyright © Manu Aranda | Fran Sánchez 2022 y Javier Mariscal | Rafa | Miguel Angel Pintado 2022

Programación de la API 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.

```
// echo $_SESSION['user']['nie'];
include('alumnnav.php');
// Creamos la consulta
$url = "http://192.168.1.96:8080/sancionactual.php?nie=".$_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'>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>
    <div>
      <?php
      foreach($data as $value){
        $fecha = $value[0];
        $fechainicio = $value[1];
        $fechafin = $value[2];
        $descripcion = $value[3];
        $tipo = $value[4];
```



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

SANCIONES ACTIVAS

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

Copyright © Manu Aranda | Fran Sánchez 2022 y Javier Mariscal | Rafa | Miguel Angel Pintado 2022

Programación de la API Shikoba

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


```
// Creamos la consulta
$url = "http://192.168.1.96:8080/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);
//var_dump($data);
//echo '<pre>',print_r($data,1),'</pre>';

?>
<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];
```



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

LISTADO DE SANCIONES

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

Copyright © Manu Aranda | Fran Sánchez 2022 y Javier Mariscal | Rafa | Miguel Angel Pintado 2022

¿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.

RECURSOS

SOFTWARE:

- Instalación de una pila LAMP en DOCKER.
- Servidor web en el cual hemos usado apache2 y PHP.
- PHPMYADMIN de forma gráfica.
- Bases de datos en MARIADB
- Programas de desarrollo: VISUAL STUDIO CODE.

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, lighter blue 3D triangle in the top left corner; a small, dark blue 3D triangle in the bottom left corner; a small, light blue 3D triangle in the bottom right corner; and a series of concentric, dark blue triangles at the bottom center.

GRACIAS POR SU
ATENCIÓN