

Por Jorge Said Serrano Soto

2021

Tiendita web
Tu mejor
opción

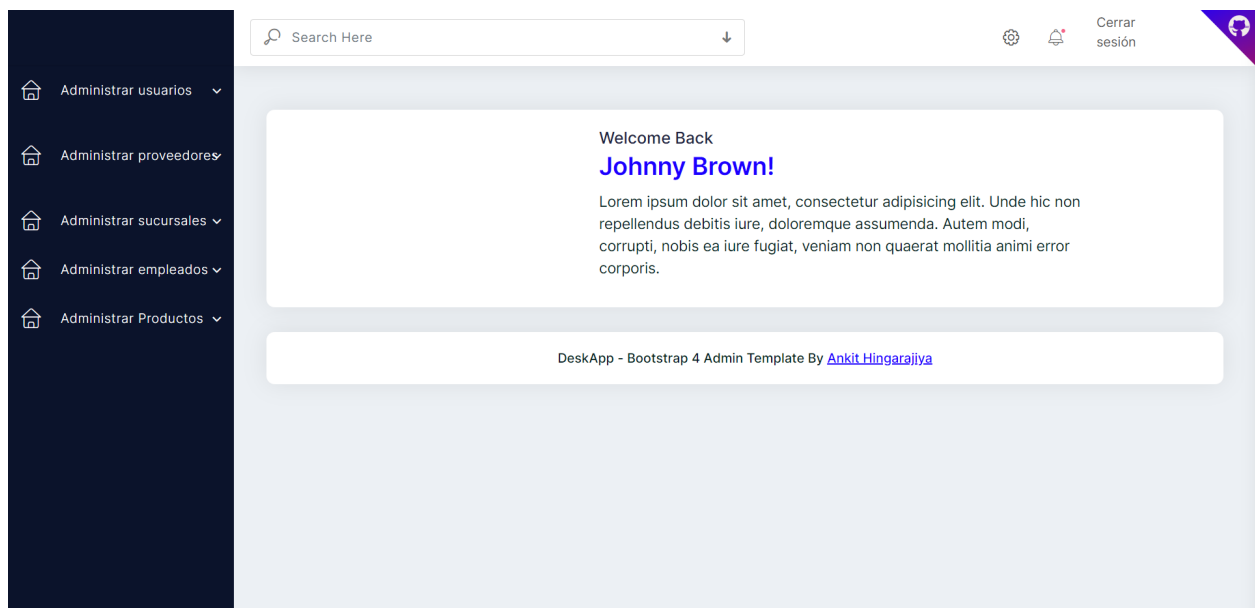
Interfaces de usuario	4
Inicio	4
Administración de usuarios	4
Administración de proveedores	5
Administración de sucursales	5
Administración de empleados	5
Administración de productos	6
Funcionalidades	6
Insertar datos referente al apartado	6
Eliminar datos según el apartado	7
Actualización de cambio de datos	7

Interfaces de usuario

Esta web está conformada por las siguientes interfaces de usuario que logran en conjunto crear un producto de administración para tiendas de abarrotes bastante eficiente.

Inicio

Esta pantalla logra dar al usuario la posibilidad de poder navegar entre las diferentes opciones para poder administrar los datos como el de los empleados, los productos o sucursales que este tiene en cuanto a su negocio se refiere.



Administración de usuarios

En esta interfaz el usuario podrá administrar a los usuarios que se encuentren dentro del sistema, con el fin de llevar un mejor control de su herramienta de administración.

Administración de usuarios

[Volver al tablero](#)

[Add](#)

Nombre	Apellido paterno	Apellido materno	Nombre de usuario	Correo	Tipo de usuario	Action
Alex	Lora	Montoya	alex4545	alex@gmail.com	1	Edit Delete

Administración de proveedores

Aquí es en donde el usuario hace la debida administración de los proveedores que se encuentran dentro del negocio con el fin de llevar un mejor control de los mismos.

Administración de proveedores

[Volver al tablero](#)

[Add](#)

Id proveedor	Descripción	Numero de proveedor		
1	Barcel	9992031886	Edit	Delete
2	Sabritas	9992031886	Edit	Delete

Administración de sucursales

Esta pantalla logra darle al usuario la posibilidad de poder verificar las sucursales que se encuentran dentro del sistema.

Administración de Sucursales

[Volver al tablero](#)

[Add](#)

Nombre de sucursal	Ubicación	administrador de sucursal	ACTION
Tendejon los caribes	calle 99b numero 950 por 45l y 45m Fraccionamiento las americanas	Alex	Edit Delete

Administración de empleados

En esta pantalla el tendero podrá ver a todos los empleados que se encuentren en todas las sucursales.

Administración de empleados

[Volver al tablero](#)

[Add](#)

id del empleado	Salario	Sucursal de empleado	ACTION	
2	56	Tendejon los caribes	Edit	Delete
3	600	Tendejon los caribes	Edit	Delete

Administración de productos

Esta pantalla funciona como panel administrador de productos para que el usuario pueda instalarlos con el fin de ver datos como el stock de inventario.

Administración de Productos

[Volver al tablero](#)[Add](#)

Nombre de producto	Precio	Stock	Proveedor	ACTION
Doritos	18	45	Sabritas	Edit Delete

Funcionalidades

Insertar datos referente al apartado

Esta funcionalidad permite al usuario insertar datos conforme el usuario así lo desee, con el fin de que posteriormente los datos se vean reflejados en el sistema, se toma en cuenta de que la inserción requiere de la captura de datos por lo tanto se le proporciona al usuario campos de captura para que pueda insertar datos.


Agregar producto

Nombre de producto

Precio

Stock

Proveedor que lo proporciona

Barcel 

Enviar

Eliminar datos según el apartado

Esta funcionalidad como lo indica su nombre permite al usuario eliminar datos que ya no puedan llegar a ser de utilidad, esto se logra presionando el botón de eliminación y dando la confirmación de eliminación de datos.

Administración de Productos

[Volver al tablero](#)

[Add](#)

[::1] dice
Are you sure You want to delete?

Aceptar

Cancelar

Nombre de producto	Precio	Stock	Proveedor	ACTION
Doritos	18	45	Sabritas	Edit Delete

Actualización de cambio de datos

Esta funcionalidad tiene como objetivo cambiar algún apartado de algún dato según la necesidad del usuario presente, para esto nos guía a un pantalla en donde los datos pueden llegar a ser cambiados de manera correctiva o con fines de actualizar información.

Producto Edit

Nombre del producto

Doritos

Precio de producto

18

Stock

45

Proveedor

Barcel

[Save](#)

Estructura del código

Estructura de controladores

```
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Empleado extends CI_Controller {
5
6      function __construct()
7      {
8          parent::__construct();
9          $this->load->library('pagination');
10         $this->load->helper('form');
11     }
12
13
14     public function index()
15     {
16         $ch = curl_init();
17         curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Empleado');
18         curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
19         curl_setopt($ch, CURLOPT_HEADER, 0);
20         $data = curl_exec($ch);
21
22         $data2= json_decode($data, true);
23
24         $data_empleado['empleados'] = $data2;
25
26         curl_close($ch);
27
28         $ch = curl_init();
29         curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Sucursal');
30         curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
31         curl_setopt($ch, CURLOPT_HEADER, 0);
32         $data = curl_exec($ch);
33
34         $data3= json_decode($data, true);
35
36         $data_empleado['sucursales'] = $data3;
37
38         $data2['pagination'] = $this->pagination->create_links();
39         $this->load->view('admin_empleados', $data_empleado);
40     }
41
42     public function loadAdd(){
43
44         $ch = curl_init();
45         curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Sucursal');
46         curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

Cada controlador que maneja la webapp se compone de cinco métodos que manejan la estructura general del funcionamiento de la webapp.

Metodo index

El método index se utiliza para redirigir a la vista principal con el fin de que se logre cargar los datos necesarios para el manejo de dicha vista,

```
public function index()
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Empleado');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    $data = curl_exec($ch);

    $data2= json_decode($data, true);

    $data_empleado['empleados']= $data2;

    curl_close($ch);

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Sucursal');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    $data = curl_exec($ch);

    $data3= json_decode($data, true);

    $data_empleado['sucursales']= $data3;

    $data2['pagination'] = $this->pagination->create_links();
    $this->load->view('admin_empleados', $data_empleado);
}
```

Metodo loadAdd y add

El método load Add se encarga de redirigir a la vista de agregación de elementos, esta utiliza parámetros de llamadas a la api según se requieran ciertos datos.

```
public function loadAdd(){

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Sucursal');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    $data = curl_exec($ch);

    $data2= json_decode($data, true);

    $data_sucursal['sucursales']= $data2;

    $this->load->view('empleado_crud/add', $data_sucursal);
}
```



```

public function add(){
    function postapi($resourcePath, $method, $body = null)
    {
        $url = "http://localhost:3000/Empleado/". $resourcePath;
        //
        $ch = curl_init($url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        if($method == "POST") {
            curl_setopt($ch, CURLOPT_POST, true);
        }

        curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type: application/json"));

        if($body != null) {
            curl_setopt($ch, CURLOPT_POSTFIELDS, $body);
        }

        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

        $data = curl_exec($ch);
        if (curl_errno($ch)) {

        } else {
            $statusCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        }

        curl_close($ch);
        if ($data != false) {

            $result = json_decode($data);

            if ($result != null) {
                return $result;
            }
        }
        return false;
    }
}

```

```

    $salario= $this->input->post("salario");
    $idSucursal= $this->input->post("idSucursal");

    $data= array();
    $data['salario']= $salario;
    $data['id_sucursal']= $idSucursal;

    postapi($idSucursal, "POST", json_encode($data));

    redirect("empleado");
}

```

Metodo delete

Funciona para eliminar ciertos elementos que se presentan en la vista.

```
public function delete($idEmpleado){
    $url = "http://localhost:3000/Empleado/".$idEmpleado;
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
    $result = curl_exec($ch);
    $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    curl_close($ch);

    redirect("empleado");
}
```

Metodo edit

Este método funciona para editar ciertos datos con fines correctivos o de actualización.

```
public function edit($idEmpleado){
    try{
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Empleado');
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_HEADER, 0);
        $data = curl_exec($ch);

        $data2= json_decode($data, true);

        $data_enviar= array();

        foreach($data2 as $empleado){
            if($empleado['id_empleado'] == $idEmpleado){
                $data_enviar['empleado']= $empleado;
                break;
            }
        }

        if(isset($data_enviar['empleado']['id_empleado'])){

            $params = array(
                'salario'=> $this->input->post('salario'),
                'id_sucursal'=> $this->input->post('idSucursal')
            );

            if(isset($_POST) && count($_POST) > 0){
                $url = 'http://localhost:3000/Empleado/'.$idEmpleado;
                $data_json = json_encode($params);

                $ch = curl_init();
                curl_setopt($ch, CURLOPT_URL, $url);
                curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/json','Content-Length: ' . strlen($data_json)));
                curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT');
                curl_setopt($ch, CURLOPT_POSTFIELDS,$data_json);
                curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
                $response = curl_exec($ch);
                curl_close($ch);
                redirect("empleado");
            }else{
```

```

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, 'http://localhost:3000/Sucursal');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    $data = curl_exec($ch);

    $data2= json_decode($data, true);

    $data_enviar['sucursales']= $data2;

    $this->load->view('empleado_crud/edit',$data_enviar);
}
}

```

Estructura de cruds

Se muestra por cada apartado del programa que hay una ventana de edición y de adición, dichos archivos son vistas que el usuario usa para fines propios.

