



# Sistema de Información Droguería Promedic El Gran C.H

- Natalia Ruiz Parra
- Sara Valentina Garcia Garcia
  - Erick Hazann Zarto Gil
- Darwin Stiven Mora Martinez
- Jhonny Santiago Espitia Mendoza

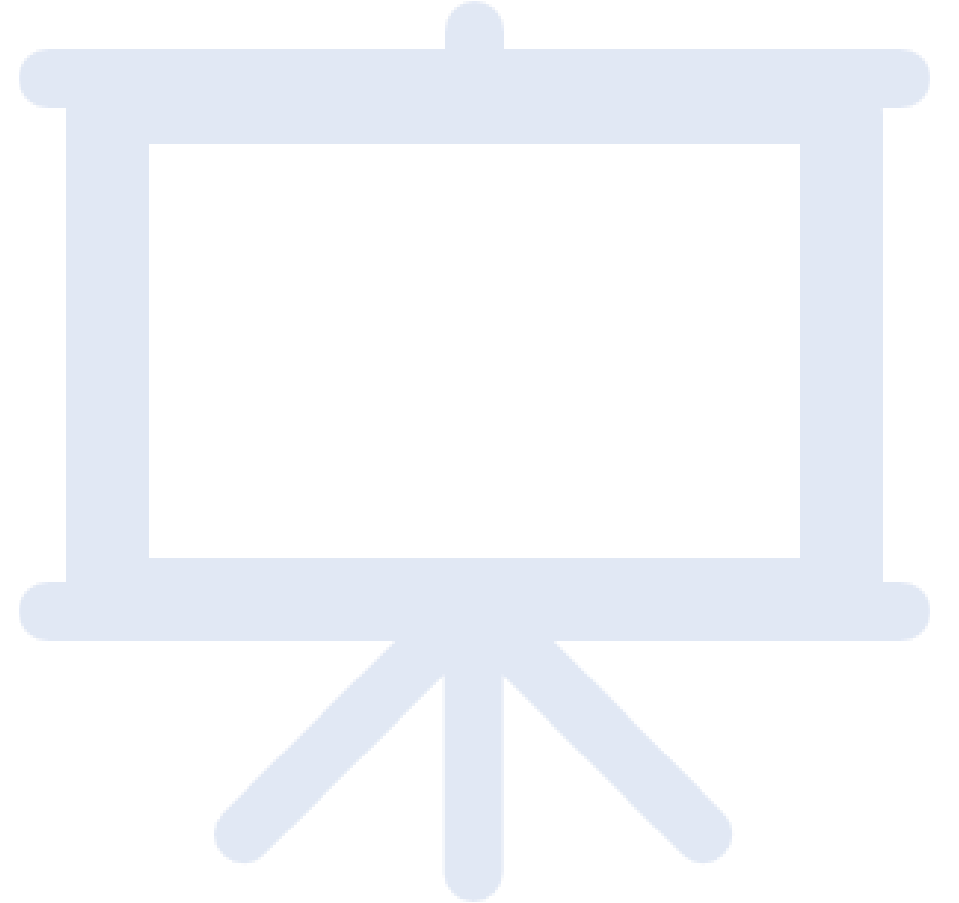
An abstract digital cityscape composed of glowing blue cubes and rectangular blocks. The structures are interconnected by a network of thin, glowing lines and dots, suggesting a data network or a digital city. Several bright blue and red light sources are scattered throughout the scene, creating a sense of depth and activity. The overall color palette is dominated by deep blues and teals, with occasional highlights of red and green.

# Índice

- 1. Presentación.
  - 1.1 Objetivo API.
- 2. Vistas
- 3. Metodos
  - 3.1 Get
  - 3.2 Post y delete
- 4. BD consumo interno.
  - 4.1 Diccionario de datos.
- 5. API externa.
- 6. Tunder y posman.



# 1. Presentación



# Objetivo API

Desarrollar una API interna que notifique a los administradores y regentes cuando un producto en inventario alcance niveles bajos de stock o se acerque a su fecha de caducidad. La API enviará alertas automáticas que incluirán información sobre el producto, su cantidad actual en inventario y la fecha de caducidad, garantizando una acción oportuna para la reposición de stock y la gestión de productos próximos a vencer.



## 2. Vistas



### Notificaciones

- El producto 'Prueba6' ha vencido.
- El producto 'Prueb2' ha superado la cantidad máxima.
- El producto 'Prueba' ha vencido.
- El producto 'Prueba' está por debajo de la cantidad mínima.
- AVISO A REGENTE: Comprar acetaminofen

AgregarCerrar



## 3.1 Metodos

Los métodos HTTP como **GET**, **POST** y **DELETE** son fundamentales para la interacción entre clientes y servidores en aplicaciones web. **GET** se utiliza para obtener datos sin modificar el estado del servidor, **POST** se emplea para enviar datos y crear o actualizar recursos en el servidor, y **DELETE** permite eliminar recursos específicos.





## 3.1 GET

Este código PHP consulta los productos en una base de datos y genera notificaciones basadas en tres condiciones: si un producto está vencido, si su stock es inferior al mínimo o superior al máximo. Si se cumple alguna de estas condiciones, se añade una notificación. Si no hay productos, se notifica que la base de datos está vacía. Al final, cierra la conexión con la base de datos y devuelve las notificaciones en formato JSON para ser procesadas por el cliente.

```
<?php
session_start();

// Conexión a la base de datos
include 'Conexion.php';

// Consulta para obtener los productos
$sql = "SELECT * FROM producto";
$result = $conn->query($sql);

$notificaciones = [];

if ($result->num_rows > 0) {
    // Procesar cada producto
    while ($row = $result->fetch_assoc()) {
        // Verificar si el producto está vencido
        if (strtotime($row['Fecha_Vencimiento']) < time()) {
            $notificaciones[] = "El producto '{$row['Nombre_Producto']}' ha vencido.";
        }
        // Verificar si está por debajo de la cantidad mínima
        if ($row['Cantidad_Stock'] < $row['Cantidad_Minima']) {
            $notificaciones[] = "El producto '{$row['Nombre_Producto']}' está por debajo de la cantidad mínima.";
        }
        // Verificar si ha alcanzado la cantidad máxima
        if ($row['Cantidad_Stock'] > $row['Cantidad_Maxima']) {
            $notificaciones[] = "El producto '{$row['Nombre_Producto']}' ha superado la cantidad máxima.";
        }
    }
} else {
    $notificaciones[] = "No hay productos en la base de datos.";
}
```

## 3.2 POST Y DELETE

```
if (!isset($_SESSION['notificaciones'])) {  
    $_SESSION['notificaciones'] = [];  
}  
  
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    // Agregar una notificación  
    if (isset($_POST['mensaje']) && !empty($_POST['mensaje'])) {  
        $mensaje = $_POST['mensaje'];  
        $_SESSION['notificaciones'][] = $mensaje; // Guardar en la sesión  
        echo json_encode(['status' => 'success', 'mensaje' => 'Notificación agregada.']);  
    } else {  
        echo json_encode(['status' => 'error', 'mensaje' => 'El mensaje no puede estar vacío.']);  
    }  
} elseif ($_SERVER['REQUEST_METHOD'] === 'DELETE') {  
    // Eliminar una notificación  
    parse_str(file_get_contents("php://input"), $_DELETE);  
    $mensaje = $_DELETE['mensaje'];  
  
    // Comprobar si el mensaje existe en las notificaciones de la sesión  
    if (($key = array_search($mensaje, $_SESSION['notificaciones'])) !== false) {  
        unset($_SESSION['notificaciones'][$key]);  
        $_SESSION['notificaciones'] = array_values($_SESSION['notificaciones']); // Reindexar la sesión  
        echo json_encode(['status' => 'success', 'mensaje' => 'Notificación eliminada.']);  
    } else {  
        echo json_encode(['status' => 'error', 'mensaje' => 'Notificación no encontrada.']);  
    }  
} else {  
    // Obtener todas las notificaciones  
    echo json_encode($_SESSION['notificaciones']);  
}
```

Este código PHP maneja notificaciones en la sesión del usuario. Primero, inicia una sesión para guardar las notificaciones. Si la solicitud es **POST**, agrega una nueva notificación a la sesión si se proporciona un mensaje. Si el mensaje está vacío, responde con un error. Si la solicitud es **DELETE**, elimina una notificación específica de la sesión, buscando el mensaje y eliminándolo si existe. Si la solicitud es de otro tipo, devuelve todas las notificaciones almacenadas en la sesión en formato JSON.





## 4. BD consumo interno

El consumo de la base de datos en este proyecto es interno porque toda la interacción con la base de datos se realiza dentro del mismo entorno (servidor local). El servidor web que ejecuta el código PHP también aloja la base de datos MySQL o similar. Esto significa que las consultas a la base de datos, como las realizadas con **GET** o **POST**, no requieren conexiones a servidores externos, sino que ocurren dentro de la misma infraestructura.

```
<?php
$conn = mysqli_connect ("localhost:3306", "root", "", "promedicch");
if(!$conn)

?>
```

## 5. API externa

Una API externa que proporciona significados sobre medicamentos permite a los usuarios acceder a información detallada y confiable sobre fármacos de manera rápida y eficiente. A través de esta herramienta, es posible realizar búsquedas específicas sobre medicamentos, obteniendo datos clave como su composición, usos, efectos secundarios, interacciones con otros fármacos, dosis recomendadas y contraindicaciones.

```
<script>

document.getElementById("searchProduct").addEventListener("click", () => {
  const productName = document.getElementById("productName").value.trim();

  if (!productName) {
    alert("Por favor, ingrese un nombre de producto.");
    return;
  }

  // Asegurarnos de que el nombre del producto esté correctamente codificado para la URL
  const encodedProductName = encodeURIComponent(productName);

  // Generar la URL de búsqueda en DrugBank
  const searchURL = "https://go.drugbank.com/unearth/q?searcher=drugs&query=${encodedProductName}&button=";

  // Abrir la página de búsqueda en DrugBank en una nueva pestaña
  window.open(searchURL, "_blank");

  // Cerrar el modal después de realizar la búsqueda
  const modal = bootstrap.Modal.getInstance(document.getElementById("searchModal"));
  modal.hide();
});
</script>
```



## 6. Posman y tunder

Codigo de la api con dependencia express.

```
1 const express = require('express');
2 const app = express();
3
4 app.use(express.json());
5
6 // Simulación de datos de notificaciones
7 const notificaciones = [
8   { id: 1, mensaje: "El producto 'Paracetamol' ha vencido." },
9   { id: 2, mensaje: "El producto 'Ibuprofeno' está por debajo de la cantidad mínima." },
10  { id: 3, mensaje: "El producto 'Aspirina' ha superado la cantidad máxima." },
11 ];
12
13 // Ruta principal
14 app.get('/', (req, res) => {
15   res.send('API de Notificaciones');
16 });
17
18 // Obtener todas las notificaciones (GET)
19 app.get('/api/notificaciones', (req, res) => {
20   res.send(notificaciones);
21 });
22
23 // Obtener una notificación específica por ID (GET)
24 app.get('/api/notificaciones/:id', (req, res) => {
```

POSTMAN

My Workspace

New HTTP Request

Filter

New Collection

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POSTMAN CONSOLE

Servidor escuchando en el puerto 80...

node

node

Lín. 42, Col. 36 (23 seleccionada) Espacios: 4 UTF-8 CRLF {} JavaScript Indents: 0 Go Live Prettier

10:05 p. m.

Get para obtener todas las notificaciones

The image shows the Postman application interface. At the top, there are tabs for "index.js" and "localhost/api/n...". The main area is titled "localhost/api/notificaciones" and shows a "GET" request. The "Body" tab is selected, displaying a JSON array of three objects. The status bar at the bottom indicates a "200 OK" response with a time of 27 ms and a size of 460 B. The terminal at the bottom shows the command "Servidor escuchando en el puerto 80...".

localhost/api/notificaciones

GET localhost/api/notificaciones

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 27 ms Size: 460 B

Pretty Raw Preview

```
1 [{"id":1,"mensaje":"El producto 'Paracetamol' ha vencido."},{
  "id":2,"mensaje":"El producto 'Ibuprofeno' está por debajo de la cantidad mínima."},
  {"id":3,"mensaje":"El producto 'Aspirina' ha superado la cantidad máxima."}]
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POSTMAN CONSOLE

Servidor escuchando en el puerto 80...



Get para obtener una notificación específica por ID

A screenshot of the Postman application interface. The top bar shows the menu (Archivo, Editar, Selección, Ver, Ir, ...) and a search bar with "PROMEDIC". The left sidebar contains icons for workspace management and a "New HTTP Request" button. The main area shows a GET request to "localhost/api/notificaciones/2". The "Body" tab is selected, showing a JSON response: [{"id":2,"mensaje":"El producto 'Ibuprofeno' está por debajo de la cantidad mínima."}]. The status bar at the bottom indicates "Status: 200 OK Time: 10 ms Size: 321 B". The bottom panel shows the "TERMINAL" tab with the text "Servidor escuchando en el puerto 80...".

Archivo Editar Selección Ver Ir ... PROMEDIC

POSTMAN JS index.js localhost/api/n... x

My Workspace localhost/api/notificaciones/2 Save No Environment

New HTTP Request GET localhost/api/notificaciones/2 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Code Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

1

Body Cookies Headers (7) Test Results Status: 200 OK Time: 10 ms Size: 321 B

Pretty Raw Preview

1 [{"id":2,"mensaje":"El producto 'Ibuprofeno' está por debajo de la cantidad mínima."}]

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POSTMAN CONSOLE

Servidor escuchando en el puerto 80...

node node Go Live



## Post para agregar una notificación



The image shows the Postman application interface. At the top, the URL bar indicates the request is to `localhost/api/notificaciones`. The method is set to **POST**. The **Body** tab is selected, showing a JSON payload: `{ "mensaje": "Levotiroxina ha vencido" }`. The **Send** button is visible. Below the request details, the **Test Results** section shows the response status: **Status: 200 OK**, **Time: 10 ms**, and **Size: 279 B**. The response body is displayed in the **Body** tab, showing the JSON: `{ "id": 4, "mensaje": "Levotiroxina ha vencido" }`. At the bottom, the **TERMINAL** tab is active, showing the command `node index.js` and the output `Servidor escuchando en el puerto 80...`.



Delete para borrar 1 mensaje por ID

localhost/api/notificaciones/2

DELETE localhost/api/notificaciones/2 **Send**

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings [Code](#) [Cookies](#)

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#) [Beautify](#)

```
1 {
2   "mensaje": "Levotiroxina ha vencido"
3 }
```

**Body** Cookies Headers (7) Test Results [Status: 200 OK](#) [Time: 18 ms](#) [Size: 321 B](#) [...](#) [+](#) [-](#)

[Pretty](#) [Raw](#) [Preview](#)

```
1 {"id":2,"mensaje":"El producto 'Ibuprofeno' está por debajo de la cantidad mínima."}
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL** PUERTOS POSTMAN CONSOLE

```
PS C:\xampp\htdocs\PROMEDIC> node index.js
Servidor escuchando en el puerto 80...
```

[node](#)  
[node](#)



JS index.js localhost/api/n... x

localhost/api/notificaciones Save No Environment

GET localhost/api/notificaciones Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Code Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "mensaje": "Levotiroxina ha vencido"
3 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 15 ms Size: 418 B

Pretty Raw Preview JSON

```
1 [{"id":1,"mensaje":"El producto 'Paracetamol' ha vencido."},{
  "id":3,"mensaje":"El producto 'Aspirina' ha superado la
  cantidad máxima."},{
  "id":4,"mensaje":"Levotiroxina ha vencido"}]
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POSTMAN CONSOLE

```
PS C:\xampp\htdocs\PROMEDIC> node index.js
Servidor escuchando en el puerto 80...
```

node node



**G R A C I A S**

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)