

# API de Gestión de Usuarios

## **Introducción:**

Esta API proporciona operaciones para la gestión de usuarios, incluyendo autenticación y administración de cuentas de usuario.

Maneja diferentes operaciones relacionadas con la autenticación de usuarios y la gestión de sesiones. Incluye funciones para validar contraseñas, manejar el inicio de sesión de usuarios, el registro de usuarios, la verificación del estado de la sesión y el listado de usuarios (con autenticación y verificaciones de permisos)

URL Base: <http://localhost/tp2V-1.0>

## **Login**

Permite a un usuario autenticarse proporcionando su nombre de usuario, contraseña y obtener acceso a su cuenta.

Endpoint: /login.php

## **Parámetros:**

username(string):Nombre de usuario o correo electrónico,  
password(string) : Contraseña

Método: POST

Respuesta:

Éxito (200): Devuelve un objeto JSON con la siguiente estructura:

```
{  
  "success": true,  
  "message": "Login exitoso",  
  "profile_image": "imagen-perfil"  
}
```

Error (400): Si las credenciales son incorrectas o faltan datos.

```
{  
  "success": false,  
  "message": "Credenciales incorrectas"  
}
```

### **Registro nuevo usuario (signup)**

Permite a los usuarios crear una nueva cuenta en el sistema.  
Registra un nuevo usuario con nombre de usuario, email y contraseña.  
Implementa la encriptación de contraseñas con password\_hash() y password\_verify() para garantizar la seguridad de los usuarios

Endpoint: /signup.php

Método HTTP: POST

Parámetros:

username (string): Nombre de usuario para la nueva cuenta.  
email (string): Dirección de correo electrónico del usuario.  
password (string): Contraseña para la nueva cuenta.

Respuestas:

exito (200):

```
{  
  "success": true,  
  "message": "Usuario creado satisfactoriamente!"  
}
```

Error(400):

```
{  
  "success": false,  
  "message": "Error al crear el usuario">  
}
```

### **Verificación de Sesión Activa**

Verifica si el usuario tiene una sesión activa.

Endpoint: /checkSession.php

Método HTTP: GET

Respuestas:

Exito (200):

```
{
```

```
"success": true,  
"message": "El usuario tiene la sesión iniciada"  
}
```

Error (400):

```
{  
  "success": false,  
  "message": "El usuario no tiene la sesión iniciada"  
}
```

**Detalles de implementación de 'checkSession':** Verifica si un usuario tiene una sesión activa comprobando la presencia de user\_id en la \$\_SESSION. Si el usuario ha iniciado sesión, recupera el estado de la sesión desde la tabla de base de datos session-user basado en el ID del usuario. Si el estado de la sesión está en 1 (indicando una sesión activa), establece el estado como exitoso con un mensaje indicando que el usuario tiene una sesión activa. De lo contrario, establece el estado como no exitoso con un mensaje indicando que el usuario no tiene una sesión activa.

### **Listar Usuarios**

Obtiene la lista de usuarios registrados en el sistema (requiere permisos adecuados).

Endpoint: listUsers.php

Metodo: GET /users

Autenticación: Utiliza el encabezado HTTP X-Username y X-Password

Respuesta:

Exitosa

```
{  
  "success": true,  
  "users": [  
    {  
      "id": 1,  
      "name": "usuario1",  
      "email": "usuario1@gmail.com"  
    },  
    {  
      "id": 2,
```

```
        "name": "usuario2",
        "email": "usuario2@gmail.com"
    }
]
}
```

Error (400)

### Seguridad:

Implementa la encriptación de contraseñas con password\_hash() y password\_verify() para garantizar la seguridad de los usuarios.

### Autenticación y Autorización

Funciones internas para autenticar usuarios y verificar permisos dentro de la aplicación

**Función getAuthenticatedUser:** Verifica las credenciales del usuario contra la base de datos y devuelve la información del usuario si la autenticación es exitosa.

### Parámetros:

connection: conexión a la base de datos.

### Headers HTTP Necesarios:

X-Username: Nombre de usuario del usuario que intenta autenticarse.

X-Password: Contraseña del usuario que intenta autenticarse.

### Retorno:

En caso de éxito, devuelve un array con la siguiente estructura:

```
{
    "success": true,
    "user": {

        "id": 1 "name": "nombre_usuario",
```

```
        "group_id": 2
    }
}
```

Si las credenciales son incorrectas o la autenticación falla, devuelve:

```
{
    "success": false,
    "message": "Autenticación fallida"
}
```

Si ocurre un error durante la ejecución de la consulta,

```
{
    "success": false,
    "message": "Error al ejecutar la consulta de autenticación"
}
```

### **checkPermission**

La función checkPermission en PHP verifica si un grupo específico tiene permisos para realizar una acción determinada.

#### Parámetros:

\$connection: Conexión a la base de datos.

\$groupId: ID del grupo para el cual se desea verificar el permiso.

\$action: Acción específica que se desea verificar si el grupo tiene permiso para realizar.

#### Retorno:

Si el grupo tiene permiso para la acción especificada, devuelve

```
{
    "success": true
}
```

Si el grupo no tiene permiso para la acción especificada, devuelve:

```
{
    "success": false,
    "message": "Privilegios insuficientes para realizar la acción:"
}
```

Si ocurre un error durante la ejecución de la consulta, devuelve:

```
{  
  "success": false,  
  "message": "Error al ejecutar la consulta de permisos"  
}
```

### **Tablas de la Base de Datos Relacionadas:**

#### **TABLA USER**

Almacena la información básica de los usuarios registrados en el sistema.

##### Campos:

- id: Identificador único del usuario (clave primaria).
- name: Nombre de usuario único.
- password: Contraseña almacenada de forma segura mediante hash.
- email: Correo electrónico del usuario único.

#### **TABLA GROUP**

Define los diferentes grupos o roles que pueden tener los usuarios en el sistema.

##### Campos:

- id: Identificador único del grupo (clave primaria).
- name: Nombre del grupo

#### **TABLA GROUP-USER**

Relaciona usuarios con grupos, permitiendo asignar múltiples usuarios a múltiples grupos.

##### Campos:

- Id: Identificador único de la relación.
- group\_id: Clave externa que referencia group.id.
- user\_id: Clave externa que referencia user.id.

#### **SESSION-USER**

Gestiona las sesiones de los usuarios activos en el sistema.

##### Campos:

- id: Identificador único de la sesión.
- estado: Estado de la sesión (activa o inactiva).
- user\_id: Clave externa que referencia user.id.

Referencias de la creación del documento:

-<https://adictosaltrabajo.com/2015/07/02/apis-rest-documentadas-y-probadas-con-api-blueprint-y-dredd/>

<https://mensagia.com/docs/documentacion-de-la-api/1025>

- <https://developer.paypal.com/docs/api/orders/v2/>