

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PRÁCTICAS SUPERVISADAS F-

CATEDRÁTICO: ING. HERMAN VELIZ

TUTOR ACADÉMICO: KEVIN SAMAYOA



MANURAL TÉCNICO

Actividad 4

LUIS PABLO CARLOS TAPIA – 202203257

DANIEL EDUARDO MARROQUÍN POLANCO – 202200011

ANGEL JOSE CARDENAS POCON – 202304063

JOSÉ SEBASTIAN PIRIR ROMERO – 202300335

CHRISTIAN GERARDO QUIÑONEZ GUZMÁN – 202209129

GUATEMALA, 18 DE SEPTIEMBRE DE 2024

Contenido

- INTRODUCCIÓN.....3
- Estructura del Código3
- ping2.....3
- getCursos.....3
- registrarUsuario.....3
- iniciarSesion.....3
- resetPass.....3
- Funciones Detalladas3
 - ping2(req, res).....3
 - getCursos(req, res)3
 - registrarUsuario(req, res)4
 - iniciarSesion(req, res)4
 - resetPass(req, res).....5
- Conclusiones5

INTRODUCCIÓN

Este manual describe un controlador de API que gestiona la interacción con la base de datos para un sistema de gestión de cursos y usuarios. Está implementado en JavaScript utilizando Node.js y una conexión a la base de datos a través de un pool de conexiones.

Estructura del Código

El código exporta varias funciones que actúan como controladores para manejar diferentes solicitudes HTTP. Las funciones principales son:

ping2: Verifica que el controlador esté funcionando.

getCursos: Recupera una lista de cursos de la base de datos.

registrarUsuario: Registra un nuevo usuario en la base de datos.

iniciarSesion: Permite a un usuario iniciar sesión en el sistema.

resetPass: Permite a un usuario restablecer su contraseña.

Funciones Detalladas

ping2(req, res)

- ❖ **Descripción:** Endpoint para verificar el estado del controlador.
- ❖ **Método HTTP:** GET
- ❖ **Respuesta:** Devuelve un mensaje de confirmación.

```
{ "mensaje": "Ping desde el controlador" }
```

getCursos(req, res)

- ❖ **Descripción:** Recupera todos los cursos de la base de datos.
- ❖ **Método HTTP:** GET
- ❖ **Respuesta:** Devuelve un array de cursos.
- ❖ **Manejo de Errores:** Si ocurre un error, se devuelve un mensaje de error y el estado 500.

```
• [ { "id": 1, "nombre": "Curso 1", ... }, { "id": 2, "nombre": "Curso 2", ... } ]
```

registrarUsuario(req, res)

- ❖ **Descripción:** Registra un nuevo usuario con los datos proporcionados.
- ❖ **Método HTTP:** POST
- ❖ **Cuerpo de la Solicitud:**

```
{  
  "Registro_academico": "12345",  
  "Nombres": "John",  
  "Apellidos": "Doe",  
  "Contrasenia": "password123",  
  "Correo_electronico": "john.doe@example.com"  
}
```

- ❖ **Respuesta Exitosa:** Mensaje de éxito y ID del usuario creado.

```
{  
  "message": "Usuario registrado exitosamente",  
  "id": 1  
}
```

- ❖ **Manejo de Errores:** Valida que todos los campos estén presentes y devuelve errores correspondientes si faltan datos o si hay un error en la base de datos.

iniciarSesion(req, res)

- ❖ **Descripción:** Permite a un usuario iniciar sesión verificando sus credenciales.
- ❖ **Método HTTP:** POST
- ❖ **Cuerpo de la Solicitud:**

```
{  
  "Registro_academico": "12345",  
  "Contrasenia": "password123"  
}
```

- ❖ **Respuesta Exitosa:** Mensaje de éxito y datos del usuario.

```
{  
  "message": "Inicio de sesión exitoso",  
  "usuario": {  
    "id": 1,  
    "nombre": "John",  
    "correo": "john.doe@example.com"  
  }  
}
```

- ❖ **Manejo de Errores:** Verifica que los campos estén presentes y si las credenciales son correctas, devolviendo un mensaje de error apropiado en caso de fallos.

resetPass(req, res)

- ❖ **Descripción:** Permite a un usuario restablecer su contraseña.

- ❖ **Método HTTP:** POST

- ❖ **Cuerpo de la Solicitud:**

```
{  
  "registroAcademico": "12345",  
  "correoElectronico": "john.doe@example.com",  
  "nuevaContraseña": "newpassword123"  
}
```

- ❖ **Respuesta Exitosa:** Mensaje de éxito.

```
{ "message": "Contraseña actualizada exitosamente" }
```

- ❖ **Manejo de Errores:** Verifica que los datos sean correctos y devuelve mensajes de error en caso de que falten datos o no se encuentre el usuario.

Conclusiones

Este controlador proporciona una interfaz para manejar operaciones básicas relacionadas con usuarios y cursos. Implementa manejo de errores adecuado para asegurar que los clientes reciban información clara sobre el estado de sus solicitudes.