

MANUAL DE USUARIO

- **CRUD Empleados:**

El manual tiene como finalidad guiar al usuario en el uso del CRUD de empleados. Su objetivo principal es permitir la administración estructurada de la información relacionada con los empleados mediante la implementación de servicios REST.

El sistema permite al usuario ejecutar las operaciones fundamentales de persistencia de datos:

- **Creación de empleados**
- **Consultar empleados**
- **Actualizar empleados**
- **Eliminación de empleados**

- **Requisitos Técnicos**

- **3.1 Software Requerido**

- Visual Studio Code o IntelliJ IDEA
- MySQL Workbench
- Cliente de pruebas para servicios REST (Postman)
- Navegador Web

- **3.2 Configuración del Servidor**

- El módulo se encuentra disponible mediante la siguiente ruta base:
<http://localhost:8080/api/empleados>

- **Especificación de Operaciones CRUD**

- **4.1 Consulta General de Empleados**

- Descripción: Permite recuperar la totalidad de registros almacenados en la base de datos correspondientes a la entidad Empleado.

- **Método HTTP**

GET

- **Endpoint**

/api/empleados

- **Parámetros**

No requiere parámetros de entrada.

- **Respuesta del Sistema**

El servidor retorna una colección de objetos JSON con la información de los empleados almacenados.

- **4.2 Registro de Empleados**

- **Descripción:**

Permite la inserción de nuevos registros dentro de la base de datos.

- **Método HTTP**

POST

- **Endpoint**

/api/empleados

- **Estructura de Datos de Entrada**

El cliente debe enviar un objeto JSON con los atributos del empleado.

- ```
{
 "nombre": "string",
 "telefono": "string",
 "direccion": "string",
 "correo": "string"
}
```

- **Validaciones**

- No se permiten campos nulos o vacíos.
- El formato del correo electrónico debe cumplir con estándares válidos.
- Los datos deben respetar los tipos definidos en el modelo.

- **Respuesta del Sistema**

Retorna el objeto creado junto con el código de estado HTTP correspondiente.

- **4.3 Actualización de Información de Empleados**

- **Descripción:**

Permite modificar los atributos de un registro existente identificado mediante su clave primaria.

- **Método HTTP**

**PUT**

- **Endpoint**

`/api/empleados/{id}`

- **Parámetros de Ruta**

| Parámetro | Tipo   | Descripción                      |
|-----------|--------|----------------------------------|
| id        | Entero | Identificador único del empleado |

- **Estructura de Datos de Entrada**

El cliente debe enviar un objeto JSON con los atributos del empleado.

- {  
 "nombre": "string",  
 "telefono": "string",  
 "direccion": "string",  
 "correo": "string"  
}

- Validaciones

- El identificador debe existir en la base de datos.
- Los datos deben cumplir las restricciones del modelo.

- Respuesta del Sistema

Retorna el objeto actualizado o un mensaje de error si el registro no existe.

- 4.4 Eliminación de Empleados

- Descripción:

Permite la eliminación permanente de un registro en la base de datos.

- **Método HTTP**

**DELETE**

- **Endpoint**

/api/empleados/{id}

**Parámetros de Ruta**

| Parámetro | Tipo   | Descripción                |
|-----------|--------|----------------------------|
| id        | Entero | Identificador del empleado |

- **Respuesta del Sistema**

Confirma la eliminación del registro o retorna un mensaje indicando que el registro no fue localizado.

- **5. Códigos de Respuesta HTTP**

| Código                    | Descripción                       |
|---------------------------|-----------------------------------|
| 200 OK                    | Operación ejecutada correctamente |
| 201 Created               | Registro creado exitosamente      |
| 400 Bad Request           | Error en los datos enviados       |
| 404 Not Found             | Registro no encontrado            |
| 500 Internal Server Error | Error interno del servidor        |

- **6. Seguridad y Control de Datos**

El sistema implementa validaciones para garantizar la integridad de los datos, evitando:

- Inserción de información incompleta

- Modificación de registros inexistentes
- Eliminación accidental de información

- **7. Buenas Prácticas de Uso**

- Verificar la consistencia de los datos antes de enviarlos al servidor.
- Confirmar el identificador del empleado antes de ejecutar operaciones de actualización o eliminación.
- Mantener copias de seguridad de la base de datos.

- **8. Consideraciones Finales**

El módulo CRUD de empleados permite la administración eficiente de la información mediante una arquitectura basada en servicios REST, facilitando la integración con otros módulos del sistema.