

DWEC – Javascript Web Cliente.

DAW Cliente - Rutas RESTful

Las rutas **RESTful** son un enfoque de diseño de API (Interfaz de Programación de Aplicaciones) que sigue los principios de REST (Transferencia de Estado Representacional). REST es un conjunto de convenciones y restricciones para diseñar servicios web que utilizan el protocolo HTTP de manera coherente y eficiente.

En el contexto de las rutas RESTful, se refiere a la forma en que los recursos (entidades o datos) son expuestos a través de las URLs (Uniform Resource Locators). Las rutas siguen un patrón jerárquico y están diseñadas de manera lógica y fácil de entender. Algunas de las características clave de las rutas RESTful incluyen:

1. **Nombres de Recursos Significativos:** Las rutas deben representar recursos específicos en lugar de acciones. Por ejemplo, en lugar de tener una ruta como **/obtenerCliente**, debería ser **/clientes**.
2. **Uso de Métodos HTTP:** Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) son mapeadas a los métodos HTTP estándar. Por ejemplo:
 - GET para obtener información.
 - POST para crear nuevos recursos.
 - PUT o PATCH para actualizar recursos.
 - DELETE para eliminar recursos.
3. **Jerarquía y Anidamiento:** La jerarquía de las rutas debe reflejar la relación entre los recursos. Por ejemplo, **/clientes/123/productos** podría representar los productos asociados al cliente con el ID 123.
4. **Uso de Plurales:** Los nombres de recursos en las rutas generalmente se escriben en plural para representar colecciones de recursos. Por ejemplo, **/clientes** en lugar de **/cliente**.

Ejemplo de rutas RESTful para un sistema de gestión de clientes:

- **GET /clientes:** Obtener la lista de todos los clientes.
- **GET /clientes/123:** Obtener detalles del cliente con ID 123.
- **POST /clientes:** Crear un nuevo cliente.
- **PUT /clientes/123:** Actualizar los detalles del cliente con ID 123.
- **DELETE /clientes/123:** Eliminar el cliente con ID 123.

Al seguir estas convenciones, se logra un diseño de API consistente, fácil de entender y que aprovecha las capacidades del protocolo HTTP de manera eficiente.