

# Unidad 6

## Servicios Web

Tecnologías de desarrollo de software IDE



# Contenidos

- ¿Qué son los Servicios Web?
- ¿Qué problemas resuelven?
- Evolución de los Servicios Web
- Tipos de Servicios Web
- Servicios Web RESTful
  - Características
  - Códigos de respuesta
- Arquitectura RESTful
- OpenAPI (anteriormente Swagger)
- Ejemplos de uso
- Implementación sobre ASP.NET Core

# ¿Qué son los Servicios Web?

Los servicios web son componentes de software que permiten que diferentes aplicaciones se comuniquen e interactúen entre sí a través de una red, generalmente Internet.

Otras aplicaciones pueden acceder a los servicios web y utilizarlos mediante tecnologías web estándar, lo que los hace independientes de la plataforma y del lenguaje de programación.

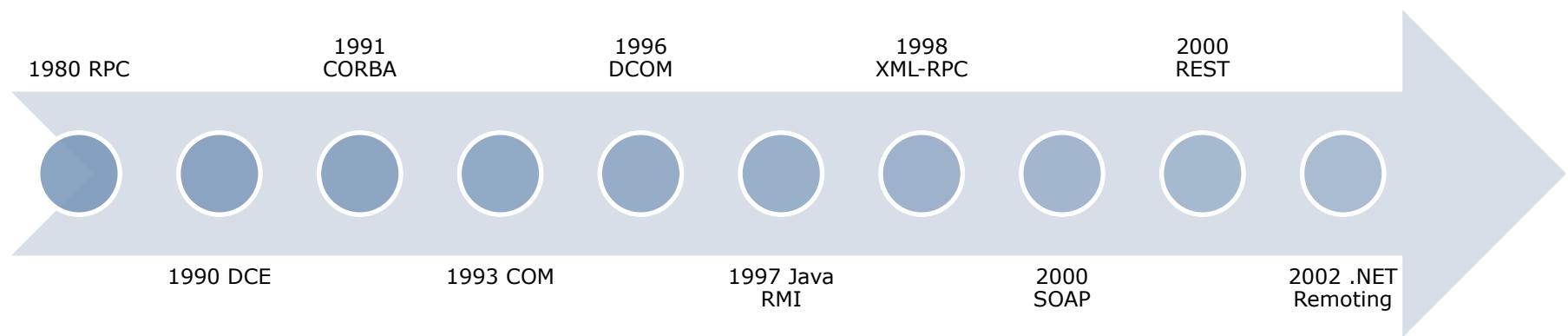
# ¿Qué problemas resuelven?

Los servicios web proporcionan un mecanismo para que las aplicaciones intercambien datos e invoquen funcionalidades específicas de forma remota.

Permiten una integración e interoperabilidad perfectas entre diferentes sistemas, independientemente de sus tecnologías subyacentes.

Juegan un papel crucial en la habilitación de la computación distribuida, permitiendo que las aplicaciones aprovechen las capacidades de las demás y compartan datos de manera segura y eficiente.

# Evolución de los Servicios Web



# Tipos de Servicios Web

- Basados en SOAP
  - Protocolo SOAP (Simple Object Access Protocol)
  - XML para intercambio de datos
- RESTful
  - Principios REST (REpresentational State Transfer)
  - HTTP + JSON o XML para intercambio

# Servicios Web RESTful

- También se los conoce como REST APIs
- Estilo arquitectónico popular para diseñar e implementar servicios web
- Se basan en un conjunto de principios que enfatizan:
  - Simplicidad
  - Escalabilidad
  - Sin estado (stateless)
- Usados ampliamente en aplicaciones modernas

# Servicios Web RESTful – Características

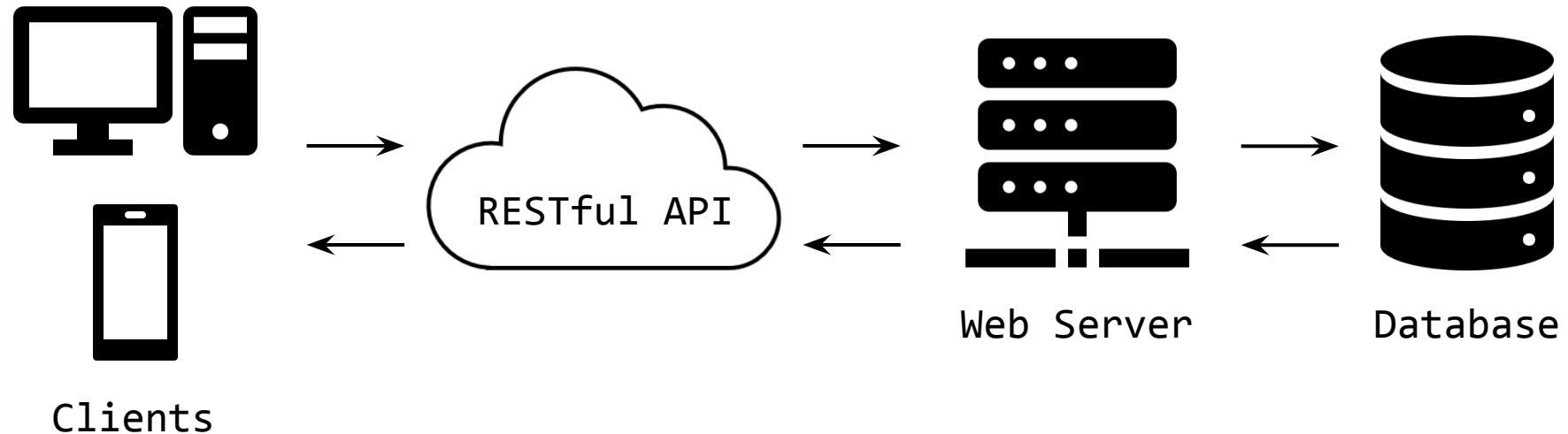
- Orientado a recursos
- Sin estado (stateless)
- Operaciones CRUD
  - GET: recuperar un recurso o una colección de recursos
  - POST: crear un nuevo recurso
  - PUT: actualizar un recurso existente
  - DELETE: eliminar un recurso
- Interfaz uniforme
- Formatos de datos
  - JSON (estándar de facto)
  - XML

# Servicios Web RESTful – Códigos de respuesta

Los códigos de respuesta HTTP nos brindan un diálogo rico entre clientes y servidores sobre el estado de una solicitud (request):

- 1xx: Informational
  - 100 Continue
- 2xx: Success
  - 200 OK
  - 201 Created
  - 204 No Content
- 3xx: Redirection
  - 301 Moved Permanently
  - 304 Not Modified
- 4xx: Client Error
  - 400 Bad Request
  - 401 Unauthorized
  - 403 Forbidden
  - 404 Not Found
- 5xx: Server Error
  - 500 Internal Server Error
  - 503 Service Unavailable

# Arquitectura RESTful



# OpenAPI (anteriormente Swagger)

Es una especificación que define un formato de descripción para APIs REST. Permite describir una API completa, incluyendo:

- Recursos disponibles
  - /usuarios
- Operaciones sobre cada recurso
  - GET /usuarios
  - POST /usuarios
- Parámetros de entrada y salida para cada operación
- Métodos de autenticación
- Información de contacto, licencia, términos de uso y otra información

# Ejemplos de uso

- Gateways de pago
- APIs del clima
- APIs de redes sociales
- Servicios Web gubernamentales
- ...

# Implementación sobre ASP.NET Core

Existen varios enfoques a la hora de implementar una Web API en ASP.NET Core:

- Apoyándonos en el patrón MVC (tradicional)
- Utilizando “minimal APIs” (nuevo)
- Explorando API endpoints (app de referencia)