Programación Orientada a Servicios
Servicios REST

Programa de
Ingeniería en Computación
UAM – Azcapotzalco

A cargo de:
Dra. Maricela Claudia Bravo Contreras
mcbc@correo.azc.uam.mx

CBI DIVISIÓN CIENCIAS BÁSICAS E INGENIERÍA
UAM - Azcapotzalco

Casa abierta al tiempo
Universidad Autónoma Metropolitana
Azcapotzalco

¿Qué es REST?

- REST (Representational State Transfer)
- Es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web.
- El término fue introducido por Roy Fielding en 2000, quien es uno de los principales autores de la especificación de HTTP.

¿Qué es un servicio REST?

- EST (REpresentation State Transfer).
- Los Servicios Web basados en REST intentan emular al protocolo HTTP o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones

Roy Fielding

- **Roy Thomas Fielding** nació en California, Estados Unidos.
- Autor principal de la especificación HTTP y una autoridad citada frecuentemente en la materia de Arquitectura de Redes.
- En su tesis doctoral titulada *Estilos Arquitecturales y el Diseño de Arquitecturas de Software basadas en Red*, describe la arquitectura REST.
- Fundador del servidor web Apache

SOAP o REST

- Muchos programadores de servicios Web están llegando a la conclusión que SOAP es demasiado complicado.
- Por lo tanto, están comenzando a utilizar Servicios Web basados en REST para mostrar cantidades de datos masivos.
- Este es el caso de grandes empresas como eBay y Google.

REST

- Conjunto de principios para el diseño de arquitecturas en red: escalabilidad, generalidad, extensibilidad, compatibilidad.
- REST se utiliza para describir a cualquier interfaz que transmite datos específicos de un domino sobre HTTP sin una capa adicional, como hace SOAP.
- REST no es un estándar, es un estilo de arquitectura.
- Aunque REST no es un estándar, está basado en estándares: HTTP, URL,.
- REST permite la representación de los recursos: XML, HTML, GIF, JPEG. Y los tipos MIME: text/xml, text/html.

Principio de REST: Escalabilidad

- Escalabilidad de la interacción con los componentes.
- La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ellos es la variedad de clientes que pueden acceder a través de la Web: estaciones de trabajo, sistemas industriales, dispositivos móviles,…

Generalidad

- Generalidad de interfaces.
- Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial.
- Esto no es del todo cierto para otras alternativas, como SOAP para los Servicios Web.

Extensibilidad

- Los clientes y servidores pueden ser puestos en funcionamiento durante años, por lo tanto deben ser capaces de entenderse con clientes actuales y viceversa.
- Diseñar un protocolo que permita la extensibilidad resulta muy complicado.
- HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de que permite crear nuevos métodos y tipos de contenido.

Compatibilidad

- Compatibilidad con componentes intermedios. Los más populares intermediaros son varios tipos de proxys, las caches, firewalls, y gateways.
- La compatibilidad con intermediarios permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas.

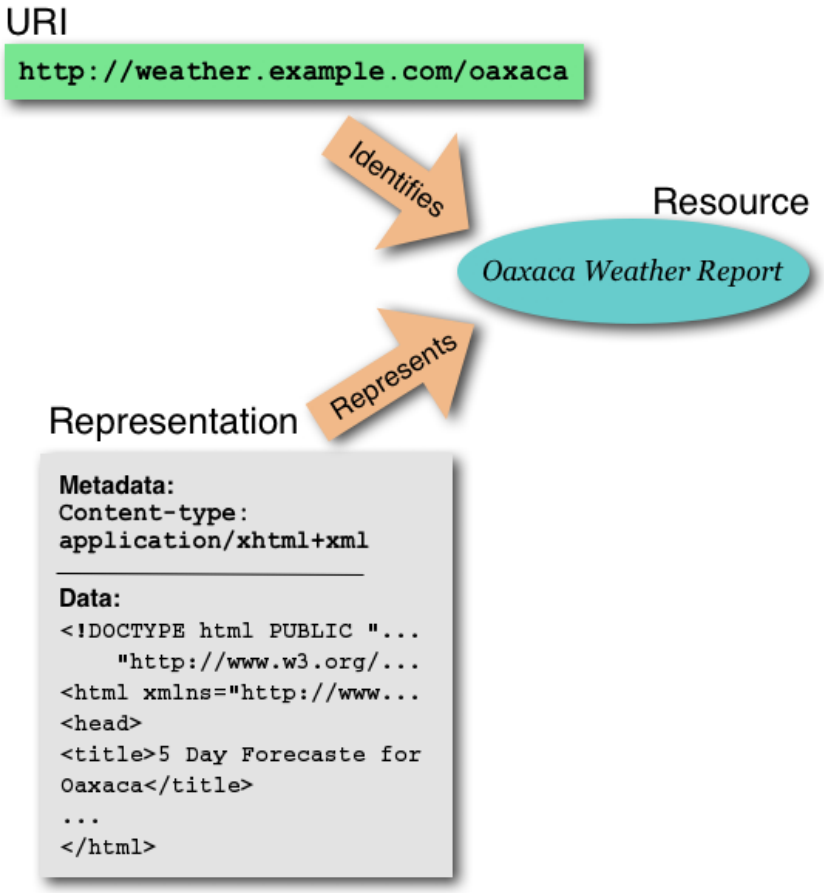
- HTTP posee un interfaz uniforme para acceso a los recursos, el cual consiste de URIs, métodos, códigos de estado, cabeceras y un contenido guiado por tipos MIME.

Métodos (operaciones) HTTP

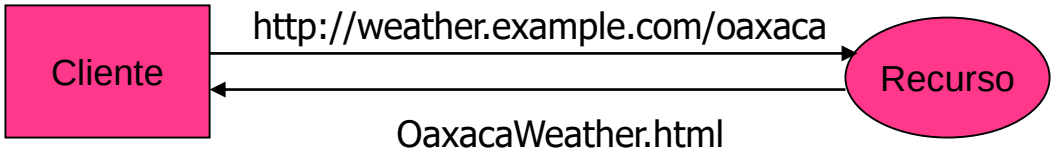
- Los métodos HTTP más importantes son:
  - POST
  - GET
  - PUT
  - DELETE.
- Operaciones asociadas a la tecnología de base de datos CRUD
  - CREATE (POST)
  - READ (GET)
  - UPDATE (PUT)
  - DELETE (DELETE)

Elementos

- Recurso
- URI-Uniform Resource Identifier (o URL)
- Web Page (HTML Page)



¿Porqué se llama Transferencia del estado de representación?



JAX-RS

JAX-RS

- Java API for RESTful Web Services es una API del lenguaje de programación Java que proporciona soporte en la creación de servicios web de acuerdo con el estilo arquitectónico Representational State Transfer (REST).
- JAX-RS usa anotaciones, introducidas en Java SE 5, para simplificar el desarrollo y despliegue de los clientes y puntos finales de los servicios web.

JAX-RS

- A partir de la versión 1.1 en adelante, JAX-RS es una parte oficial de Java EE 6.
- Una característica notable de ser parte oficial de Java EE es que no se requiere configuración para comenzar a usar JAX-RS.
- Para los entornos que no son Java EE 6 se requiere una (pequeña) entrada en el descriptor de despliegue web.xml.

Anotaciones JAX-RS

JAX-RS proporciona anotaciones para ayudar a mapear una clase recurso como un recurso web:

@Path especifica la ruta de acceso relativa para una clase recurso o método.

@GET, @PUT, @POST, @DELETE y @HEAD especifican el tipo de petición HTTP de un recurso.

@Produces especifica los tipos de medios MIME de respuesta.

@Consumes especifica los tipos de medios de petición aceptados.

Anotaciones adicionales

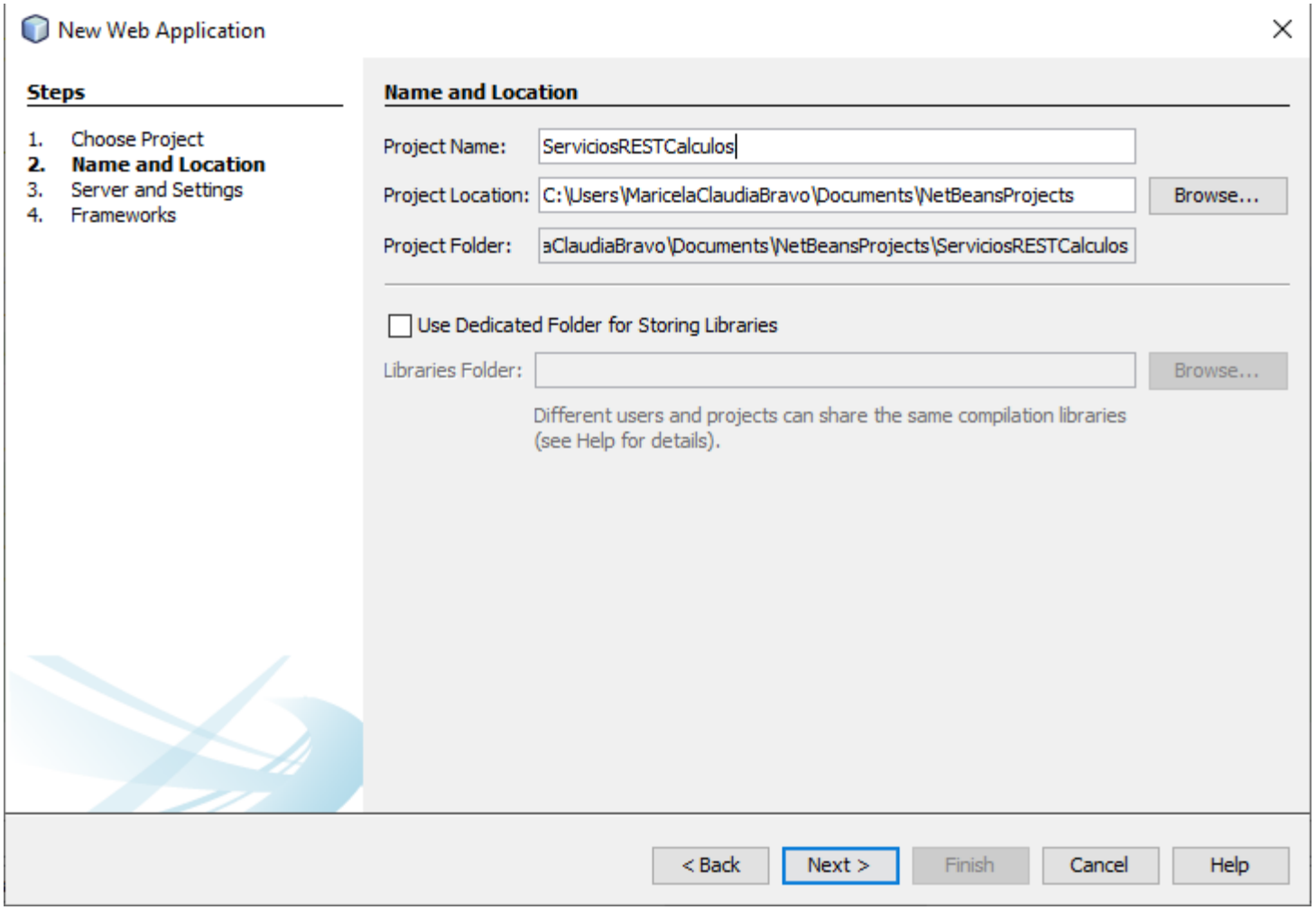
- Todas las anotaciones @*Param toman una clave de alguna forma que se utiliza para buscar el valor requerido.
  - @PathParam enlaza el parámetro a un segmento de ruta.
  - @QueryParam enlaza el parámetro al valor de un parámetro de consulta HTTP.
  - @MatrixParam enlaza el parámetro al valor de un parámetro de matriz de HTTP.
  - @HeaderParam enlaza el parámetro a un valor de cabecera HTTP.
  - @CookieParam enlaza el parámetro a un valor de cookie.
  - @FormParam enlaza el parámetro a un valor de formulario.
  - @DefaultValue especifica un valor por defecto para los enlaces anteriores cuando la clave no es encontrada.
  - @Context devuelve todo el contexto del objeto. (Por ejemplo: @Context HttpServletRequest request)

Implementación de JAX-RS

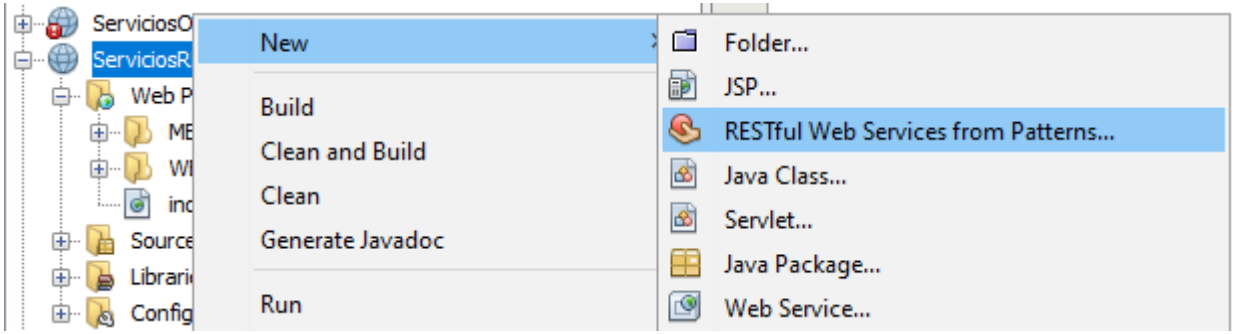
- Apache CXF, un framework de servicios web de código abierto.
- Jersey, la implementación de referencia de Sun (ahora Oracle).
- RESTeasy, implementación de JBoss.
- Restlet, creado por Jerome Louvel, un pionero en frameworks de REST.
- Apache Wink, proyecto de Apache Software Foundation Incubator, el módulo del servidor implementa JAX-RS.
- JBoss, la librería org.jboss.resteasy da soporte de JAX-RS sobre la plataforma JBoss.

EJEMPLO

1. Crear una aplicación Web



2. Craer un RESTful Web Service desde Patterns

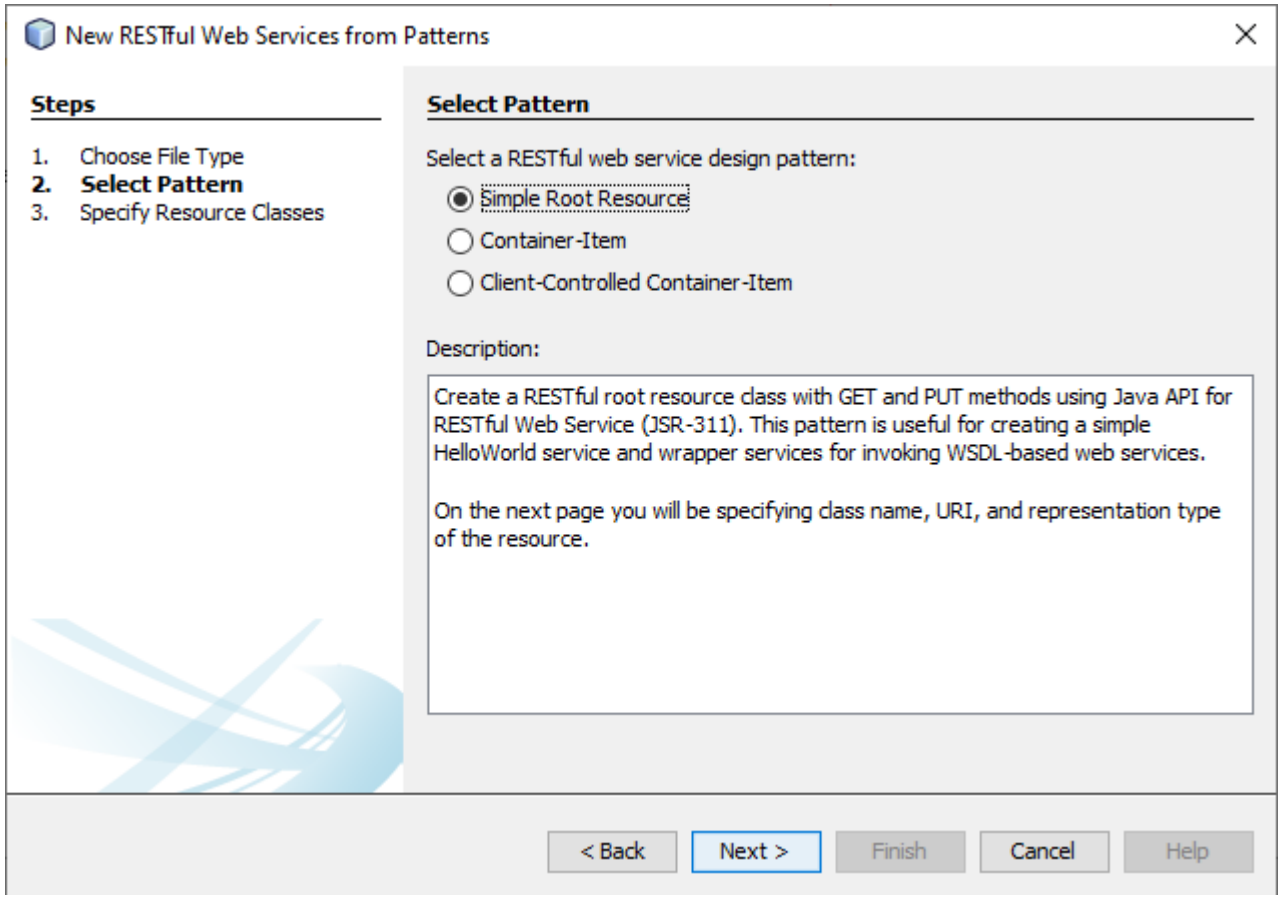


Simple Root Resource

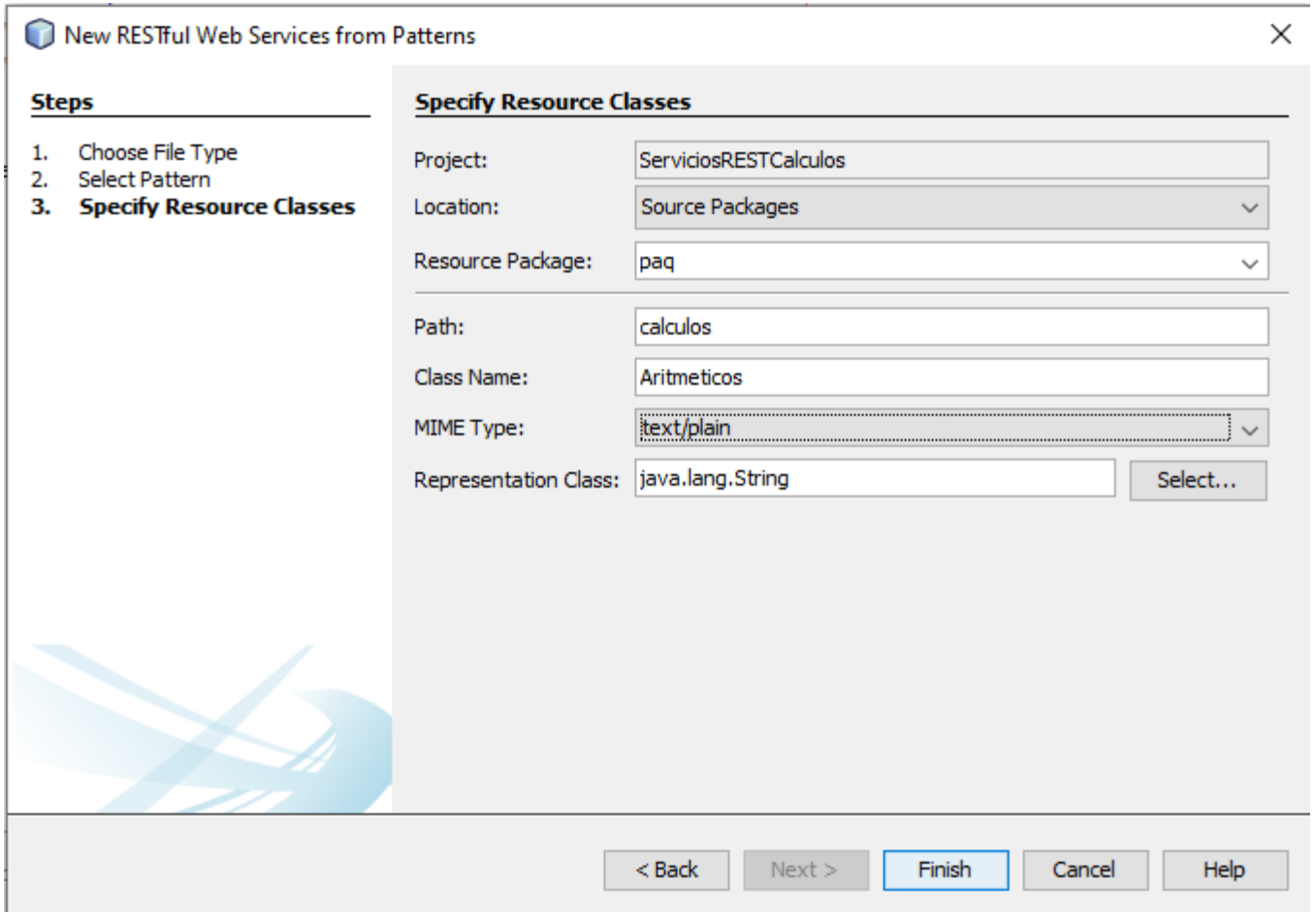
- Las clases de recursos raíz son POJO que se anotan con @Path o tienen al menos un método anotado con @Path o un designador de método de solicitud, como
  - @GET,
  - @PUT,
  - @POST o
  - @DELETE.

- JAX-RS es un lenguaje de programación Java API diseñado para facilitar el desarrollo de aplicaciones que utilizan la arquitectura REST.

3. Crear un Simple Root Resource



4. Especificar los recursos



5. Reemplazar el método GET

```
@GET
    @Produces(MediaType.TEXT_HTML)
    public String factorial(@QueryParam("base") long base)
   {
        return Long.toString($factorial(base));
    }

    long $factorial(long base) {
        if (base >= 1) {
            return $factorial(base - 1) * base;
        }
        return 1;
    }
```

6. Realizar el deploy de la aplicación Web y probar el servicio

Probar con

http://localhost:8085/ServiciosRESTCalculos/webresources/calculos?base=3

7. Crear un cliente Web

```
<html>
    <head>
        <title></title>
        <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js">
        </script>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <h2>Calcular factorial</h2>
        Número:<input type="text" name="base" id="base"/>
        <button type="button" id="calcularBtn">Calcular</button>
        <div id="resultado">
            Resultado: <span></span>
        </div>
        <script type="text/javascript">
            jQuery("#calcularBtn").click(function(){
                var base = jQuery("#base").val();
                jQuery.get("http://localhost:8085/PruebaWS1/webresources/calculos",{
                    base:base
                },function(resultado){
                    jQuery("#resultado span").text(resultado)
                })
            })
        </script>
    </body>
```