

El manejo de REST URL y sus formatos es algo con lo que nos vamos a encontrar nada más empezar a trabajar con recursos REST. ¿Cómo debemos construir una url REST? Vamos a ver un ejemplo sencillo hablando de la URL de Facturas. Si diseñamos un recurso que vaya a gestionar Facturas lo primero que necesitamos es saber en que URL se va a publicar . En nuestro caso la URL será /facturas

Colección de elementos

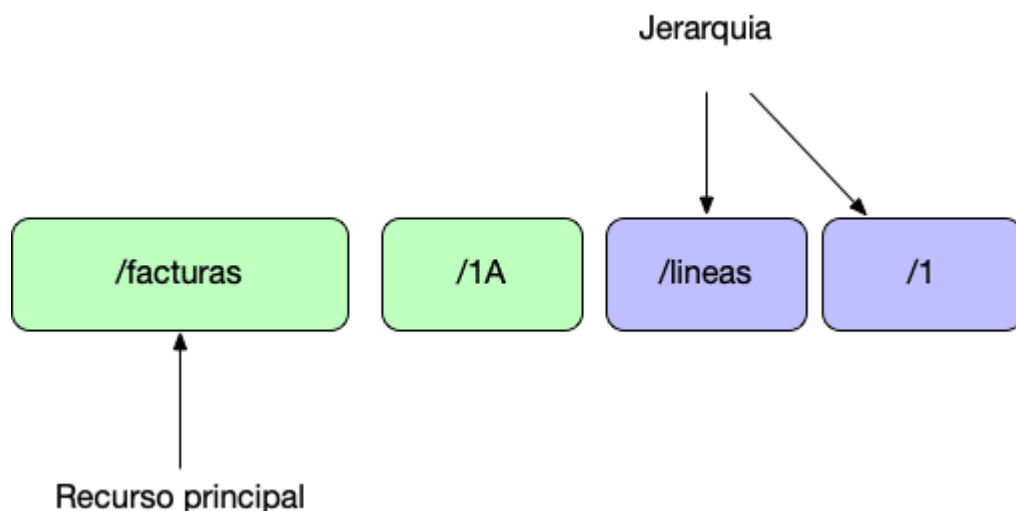
/facturas

arquitecturajava

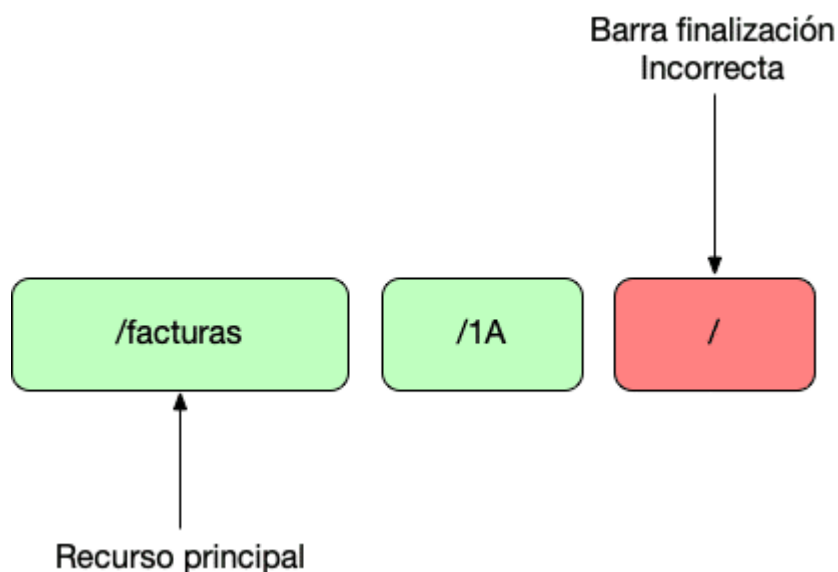
**Curso**  
**GRATIS**  
**SPRING**  
**REST**  
**Apúntate !!**

## REST URL y reglas fundamentales

1. Una colección de recursos siempre se publica en plural y en minúsculas por lo tanto /facturas es muy correcto.
2. La colección siempre hace referencia a nombres a la hora de hablar de recursos. Los verbos quedan limitados a acciones muy concretas. Es decir no tendremos una colección de /facturables sino de /facturas.
3. Los recursos siempre disponen de una estructura jerárquica de tal forma que si la factura contiene líneas una estructura razonable seria /facturas/1/lineas/1



4. Nunca usar barras de finalización . Es decir si la colección de recursos se accede con `/personas` no añadir al final una nueva barra `/personas/` salvo que existan recursos anidados.



5. Uso de guiones para clarificar el contenido de las urls. Por ejemplo nos podemos encontrar con una url de `/facturasregioncantabria` es mucho mejor rediseñarla como `/facturas-region-cantabria` ya que añade claridad al contenido .Para ello siempre usar guiones y nunca guiones bajos `/facturas_region_cantabria` sería erróneo.

`/facturas-region-cantabria`

↑  
Recurso principal  
Clarificación guiones!

6. No incluir extensiones de ficheros en las urls y delegar en las cabeceras de Content-Type para devolver un fichero pdf o excel. Por lo tanto una url como `/facturas.pdf` no se recomienda.

arquitecturajava

**Curso**  
**GRATIS**  
**SPRING**  
**REST**  
**Apúntate !!**

## Otros Artículos relacionados

- [Diseño API REST , rendimiento y OverFetching](#)
- [REST Resources Nombres vs Verbos](#)
- [Spring WebFlux Router y REST](#)
- [REST Nested Resources y su utilidad](#)