

23/02/09 - REST: Representational State Transfer

Después de haber usado durante unos años sistemas RPC para la compartición de la información, XML-RPC, SOAP y Elm; llego a REST, un concepto que mencionó primero un compañero de trabajo, Juanse, y después vi en profundidad en un curso de Ruby on Rails que se organizó en la empresa en la que trabajo (gracias Dani por ese curso tan completo).

El sistema REST se basa en usar las entidades de datos como recursos. Por ejemplo, si tenemos usuarios y queremos hacer un sistema de gestión de usuarios, mediante una URL, podemos establecer el recurso como:

```
http://bosqueviejo.net/users
```

Este recurso será accedido por HTTP, mediante el método GET, lo cual retornará, en formato serializado (XML, JSON, YAML...) la información solicitada, el listado de usuarios. También podemos especificar, mediante la agregación de un identificador, en la URL, que solo queremos un usuario concreto:

```
http://bosqueviejo.net/users/admin
```

Como se puede apreciar, el sistema de localización, es claro y limpio.

Lo más interesante viene en las siguientes acciones. Para hacer una agregación a la lista de usuarios, es decir, para insertar o agregar un nuevo usuario, se puede emplear la misma URL, agregando los datos como cuerpo del mensaje HTTP y usando el método PUT.

Por ahora, no tenemos novedades con respecto al uso estándar de los sistemas web, ya que la petición de páginas se realiza mediante GET y el envío de formularios a través de POST. Sin embargo, para las acciones de inserción, por ejemplo, se emplea PUT, con la misma dinámica que POST, solo que reacciona como una actualización sobre los datos existentes, por lo que, la URL que se usará, deberá de ser de la segunda forma que se vió anteriormente.

Por último, para eliminar un dato, usando la URL concreta como en el caso de PUT y POST, pero empleando el método DELETE, se indicará al sistema REST que se desea eliminar el recurso solicitado.

Las **ventajas** que ofrece este sistema son claras. Al hacer uso de los métodos de HTTP, el servidor web puede ser usado a nivel de autenticación para restringir en modo *todo o nada* la modificación, eliminación y agregación de recursos a usuarios no autenticados o a ciertos usuarios, e incluso, en concordancia con la raíz de URL, limitar el acceso a ciertos recursos, con ciertos métodos, etc. Además, a nivel de log del sistema web, se tienen datos más concisos sobre el acceso y uso del sistema.

Los **inconvenientes** son que las acciones tienen que ser muy específicas y, para las cadenas de búsqueda, se hace necesario el uso de los parámetros de URL, lo cual hace que no quede tan claro y limpio el sistema.

Este sistema, bien usado, constituye un punto de partida muy válido para la comunicación con sistemas BPM y la instauración de sistemas de consulta tan potentes y configurables como SQL.