

Rejunte um *middleware* para agregação e redistribuição de acervos culturais digitais na Web 2.0

Marcelo Soares Souza¹

¹Pontão de Cultura Digital juntaDados
Universidade do Estado da Bahia (UNEB) – Salvador, BA – Brasil

marcelo@juntadados.org

Abstract. *This article describes a middleware tool for the integration of audiovisual collections stored and distributed on digital platforms on the principles of Web 2.0. This solution provides an extensible architecture that provides a Web API (Application Programming Interface), based on a REST style architecture, which enables content aggregation and redistribution of content on various platforms*

Resumo. *Este artigo descreve uma ferramenta middleware para a integração de acervos de produções audiovisuais digitais armazenados e distribuídos em plataformas sobre os princípios da Web 2.0. Esta solução apresenta uma arquitetura extensível e que disponibiliza uma Web API (Application Programming Interface) baseada em uma arquitetura do estilo REST que possibilita a agregação e redistribuição de conteúdo em diversas plataformas de conteúdo na Web.*

1. Introdução

O registro e a disponibilização de produções audiovisuais em meio digital, principalmente na Web, cresceu imensamente na última década. Este fato se dá principalmente devido ao barateamento dos computadores pessoais aliado à evolução de aplicativos livres e de serviços gratuitos na rede. Estes foram os principais fatores que ajudaram a promover uma ampla democratização do acesso aos novos meios de produção e distribuição digital [CARVALHO 2009].

A existência de inúmeras plataformas para o armazenamento e distribuição de acervos culturais digitais na web permitiram uma maior difusão destas produções, disponíveis em grande parte na forma de conteúdo audiovisual [RANGEL 2010]. Porém, esta grande diversidade de plataformas dificulta a unificação e a integração de acervos digitais por meio de agregações e/ou redistribuições para novas plataformas web.

Plataformas de acervo como *Youtube*, *Flickr*, *MySpace*, *juntaDados.org* entre outras recebem milhares de arquivos de conteúdo audiovisual (áudio, imagem, vídeo, texto e etc.) diariamente e disponibilizam estes conteúdos de variadas formas. Pequenos Sites, Blogs, *Wikis* e Plataformas gerenciadoras de conteúdo surgem a todo instante e em uma Web cada vez mais dinâmica e compartilhada estas novas plataformas buscam se integrar agregando conteúdos audiovisuais já disponíveis de maneira transparente e rápida.

Na ausência de uma solução integrada e simples, este trabalho apresenta um *middleware* livre, independente de plataforma ou linguagem de programação, extensível que permite uma fácil integração de plataformas web tais como sites, portais, blogs e acervos que trabalhem com acervos audiovisuais e que compartilhem de alguma forma estes conteúdos seguindo os princípios da Web 2.0.

Este *middleware* tem como objetivo permitir um fluxo mais eficiente e eficaz na troca de conteúdos em acervos digitais, de maneira simplificada através da disponibilização de um meio e uma interface única e comum para a troca de dados (acervos).

O *middleware* Rejunte é uma solução genérica e expansível que permite a agregação de diversos conteúdos na web, bastando o desenvolvimento de *plugins* que possibilitem a extensão de suas funcionalidades de intermediação entre meios e acervos digitais.

Rejunte atua intermediando o acesso entre plataformas web que armazenem conteúdo audiovisual. Uma novo blog, site ou plataforma que necessite buscar e agregar conteúdo de outras plataformas podem através do Rejunte ter acesso simplificado a todas estas plataformas existentes.

Atualmente existem diversos meios de acesso e compartilhamento, entre plataformas, de conteúdo em plataformas web em conformidade com os princípios Web 2.0. Porém, para cada plataforma, é preciso conhecer e desenvolver os meios de acesso a cada uma das *Web API* disponíveis. O Rejunte simplifica fornecendo uma *Web API* e uma estrutura de dados única para a troca de acervos, intermediando este acesso com as diversas plataformas já existentes.

2. Compartilhamento na Web

2.1. Web 2.0

Web 2.0 é um termo cunhado por Darcy DiNucci no seu artigo *Fragmented Future*, neste artigo DiNucci discorre sobre o futuro das tecnologias e novas formas de uso da Web [RUIZ 2008].

Porém, o conceito de Web 2.0 somente começou a se tornar amplamente conhecido e debatido a partir de uma conferencia proposta por Tim O'Reilly em 2004. O'Reilly propôs a Web como uma plataforma de desenvolvimento completa como alternativa as plataformas convencionais baseadas em *Desktop* [OREILLY 2004].

Web 2.0 é um conjunto de princípios e práticas que une um conjunto de sites e plataformas que demonstram todos estes princípios propostos em ação. Web 2.0 não se refere a um padrão ou uma ferramenta específica e sim a uma mudança cumulativa na forma que a Web é utilizada por usuários e desenvolvedores [OREILLY 2009]. Web 2.0 é uma revolução nos negócios da indústria da computação causada pela mudança da Internet como plataforma e uma tentativa de entender as novas regras para o sucesso nesta nova plataforma [OREILLY 2004].

Os alicerces da Web 2.0 são as mesmas que ao que agora pode se chamar de Web 1.0, porém avanços nas formas e meios de utilização e desenvolvimento destas

mesmas bases representam a Web 2.0.

Plataformas Digitais de conteúdo baseado nos princípios da Web 2.0 já estão amplamente disponíveis. Plataformas de acervo digital como *Youtube* (Vídeo), *Flickr* (Foto/Imagem), *Twitter* (*Microblog*) dentre outros disponibilizam meios e tecnologias para o acesso, manipulação e distribuição de seus repositórios de dados [OREILLY 2009].

Em geral cada uma destas plataformas oferecem um conjunto de *Web APIs* e um formato de troca de mensagem diferente. Muitas tecnologias e técnicas permitem que estas plataformas exponham (distribuam) seus dados, geralmente através de uma *Web API* baseado em *Web Services* que permite a disponibilização dos acervos digitais a outras plataformas e sistema na web. Rejunte é baseado RESTful Web Services.

2.2. REST

REST é um estilo de arquitetura de software para Web, é também uma maneira de distribuir conteúdo de hipermídia baseado na forma como a Web esta estruturada [PAUTASSO 2008].

Originalmente *REST* surgiu como um meio de avaliar arquiteturas Web, conforme foi descrito na tese de doutorado de Roy Fielding [RICHARDSON 2007]. Fielding é um dos criadores da versão 1.1 do protocolo *HTTP* (*Hypertext Transfer Protocol*) padrão de protocolo de comunicação na camada de aplicação [RFC2616 2010]. Este estilo de arquitetura foi utilizado para explicar a excelente escalabilidade do protocolo *HTTP* 1.0 e também auxiliou no desenho da sua versão seguinte o *HTTP* 1.1 [PAUTASSO 2008].

Uma arquitetura do estilo *REST*, assim como a Web, consiste na forma clássica de comunicação cliente-servidor. No *REST*, um cliente faz a requisição de um recurso ao servidor que retorna uma resposta na forma de uma representação. Um recurso é uma fonte de informação, e pode ser qualquer coisa um vídeo, uma imagem, um documento, uma plataforma web dentre outros. A representação é o conteúdo propriamente dito deste recurso, a representação é um estado do recurso em determinado momento [PAUTASSO 2008].

Uma *Web API* baseado em *REST* é simplesmente um *Web Service* implementado sobre o protocolo *HTTP* usando os princípios do *REST*. *Web API REST* é uma coleção de recursos (serviço) que disponibiliza a sua representação (dados) [PAUTASSO 2008].

Os princípios básicos do *REST* são compostos pela identificação de recurso onde cada recurso deve ser identificado na requisição usando a *URI* (*Uniform Resource Identifier*); a manipulação do recurso que ocorre através de suas representações, quando um cliente possui a representação de um dado recurso e este pode consultar, criar, modificar ou apagar o recurso, se ele possuir a permissão suficiente para realizar tais operações; as mensagens devem ser auto descritivas ou seja cada mensagem contém informações suficientes para descrever como processar esta mensagem [RICHARDSON 2007].

O recurso requisitado é identificado através da *URI*. Este recurso pode ser

manipulado através de operações comuns ao protocolo *HTTP* (*PUT*, *GET*, *POST* e *DELETE*).

3. Middleware Rejunte

Rejunte é um *middleware* livre, modular e facilmente extensível através de *plugins*. Este funciona como uma plataforma intermediária entre plataformas web. Para tal fim disponibiliza para o cliente web uma *Web API* baseada em *Web Service* sobre uma arquitetura *REST* que possibilita a distribuição do conteúdo coletado e agregado de diversas plataformas de acervo digital.

Na atual implementação do Rejunte, a *Web API* pode ser consultada através da identificação do recurso desejado pela *URI*, ou seja, diretamente pelo endereçamento comumente utilizado na navegação Web sobre o protocolo *HTTP*. Pelo menos dois parâmetros devem ser informados através da *URI*, um que identifica o recurso (*resource*) desejado e o que identifica o critério de busca no recurso (serviço) escolhido. Também é possível, opcionalmente, informar na *URI* a quantidade máxima de registros que serão retornados. Existe a possibilidade, através da *Web API*, de se consultar a lista de recursos disponíveis no Rejunte para a posterior consultar destes.

O uso de uma arquitetura do estilo *REST* se justifica na medida que esta apresenta uma grande robustez, simplicidade no desenvolvimento e pela ampla adoção por muitas plataformas de acervos tais como *Youtube* e *Flickr*.

Nesta solução proposta, um cliente pode solicitar ao Rejunte um recurso de uma dada plataforma web e obter uma resposta em diversos formatos de troca de dados comuns na Internet tais como *POX* (*Plain Old XML*), *JSON* e *CSV* (*comma-separated values*). Por padrão, o retorno é no formato *POX*, para se obter uma resposta em outro formato é necessário informar pela *URI* o formato desejado.

A disponibilização de um novo recurso, no caso do Rejunte o equivalente a um novo acervo digital na web, é feita criando-se um novo *plugin* que implemente a forma de acesso a este novo acervo. O Rejunte age então como um intermediário entre as diversas plataformas web existentes e que disponibilizem seus acervos através de métodos diversos sobre os princípios da Web 2.0.

Detalhes de como acessar, buscar e trazer determinado conteúdo dentro de uma plataforma de acervo são irrelevantes para o *middleware* em si. O *plugin* responsável pelo acesso a uma dada plataforma web precisa apenas retornar ao *middleware* uma *array* contendo informações da mídia consultada tais como título, descrição, *URI* do arquivo da mídia, autor(es) e licença de distribuição. O *middleware* irá reformatar estes dados e expor ao cliente para que este possa consumir a representação deste recurso.

Uma aplicação Web pode consultar, por exemplo, um acervo de vídeo na plataforma *Youtube* apenas identificando o recurso e informando o parâmetro de busca através da *URI* e a resposta se dar através da representação (dados) no formato escolhido (*POX*, *JSON* e/ou *CSV*) que também pode ser informado na própria *URI*.

O Rejunte faz uso apenas do comando *GET* (sobre o protocolo *HTTP*), nenhuma informação é criada ou atualizada, como é possível na arquitetura *REST*, nos

repositórios ou no próprio Rejunte. O Rejunte tem como objetivo apenas consultar outras plataformas retornando como resultado um formato de mensagem comum.

3.1. Implementação Rejunte

Rejunte foi implementado utilizando o *framework Codeigniter* com a linguagem de programação *PHP5*. *Codeigniter* é considerado um dos melhores *framework* para desenvolvimento *PHP*, é aderente a padrões de projetos tais como *MVC (Model-View-Controller)* e *Active Record*, é extensível através de *plugins*, possui um vasta base instalado e um solido grupo de desenvolvedores [BLANCO 2009].

O Rejunte é baseado em uma arquitetura modular baseado em *plugins* que permite a adição de novas plataformas de conteúdo (acervo) de forma transparente. O suporte nativo de *plugins* no *Codeigniter* possibilitou um esforço mínimo para a elaboração desta funcionalidade.

O Rejunte disponibiliza representações dos recursos, que são referenciados através de uma simples *URI*, para que aplicações ou até mesmo o navegador (cliente web universal) acesse os dados de acervos. Para fim de avaliação foram desenvolvidos quatro *plugins* que fazem acesso a quatro plataformas distintas de armazenamento de acervos audiovisuais.

Cada plataforma web possui uma forma de disponibilizar o seu conteúdo, por exemplo: *Youtube* disponibiliza uma *Web API* baseado em *REST*, *FlickrR* possibilita acesso utilizando uma *API REST-RPC*, *Estúdio Livre* e *Portal juntaDados.org* disponibilizam seus acervos através de *feed RSS 2.0*.

Cada um dos *plugins* desenvolvidos dentro do Rejunte implementa tarefas tais como busca e organização (ordenamento) de conteúdo em cada uma destas plataformas. Os dados retornados devem ser estruturado na forma de um *array* simples e retornado ao Rejunte. O *plugin* é acionado quando o cliente consulta o Rejunte através da *URI*, informando o recurso (plataforma) desejado e a forma de busca do conteúdo desejado (por exemplo uma busca por *tags*).

O *plugin* implementa a forma de acesso ao acervo desejado e retorna ao *middleware* um *array* de dados que irá se convertido em um formato único padrão a ser disponibilizado como a representação do recurso requisitado. Um *plugin* pode ser adicionado sem necessidade de se modificar código ou reconfigurar o Rejunte, toda a operação se da de forma transparente ao cliente e em tempo real.

4. Conclusões

O *middleware* Rejunte busca preencher a lacuna de uma ferramenta web que facilite a agregação de conteúdos em plataformas e acervos digitais diversos. O *rejunte* facilita na medida em que apresenta uma interface única para o acesso a acervos digitais em plataformas web 2.0.

A existência de muitas plataformas de conteúdos digitais, e de diversas formas de acesso a estas, dificulta o desenvolvimento de uma plataforma que acesse a todos estes conteúdos e agregue estes dados de maneira simples e rápida.

Tradicionalmente, caso um desenvolvedor de uma dada solução Web precise agregar dados em diversas outras plataformas web, este deve programar o acesso para cada plataforma existente. Utilizando o Rejunte como intermediário ele apenas precisa acessar a uma única plataforma para ter acesso às diversas plataformas existentes.

O ganho primeiramente notado é o de simplificar a tarefa de busca e agregação de conteúdos em plataformas diversas na Internet. O Rejunte traz uma única interface e um único meio de acesso a diversas plataformas Web 2.0. Com o Rejunte o desenvolvedor web precisa apenas conhecer uma única *Web API* e um único formato de mensagem para ter acesso a diversas plataformas web de acervo digital, diminuindo seu trabalho consideravelmente.

Havendo a necessidade de acesso a uma nova plataforma web, ainda não disponível no Rejunte, basta o desenvolvedor criar um novo *plugin*, seguindo os exemplos disponíveis no código-fonte distribuído, para aquela nova plataforma e submeter aos mantenedores do Rejunte.

O Rejunte propõe uma ferramenta facilmente extensível que permite que novas plataformas e sites de conteúdos se preocupem apenas em acessar um único meio (*middleware*) para obter conteúdo em acervos digitais diversos e que estejam sobre os princípios da Web 2.0.

Referências

- CARVALHO JR., JOSÉ MURILO (2009). Uma ‘Cultura Digital Participativa’. Disponível em: <culturadigital.br/o-programa/uma-cultura-digital-participativa/>. Acesso em: 05 Agosto de 2010.
- RANGEL, LIA (2010). Desafios para a digitalização. Em: <culturadigital.br/simposioacervosdigitais/2010/04/26/pesquisa-em-instituicoes-vinculadas-ao-minc-aponta-desafios-para-a-digitalizacao/>. Acesso em: 05/08/2010.
- RUIZ, CARLOS (2008). Who coined Web 2.0?. Em: <www.cole20.com/who-coined-web-20-darcy-dinucci/>. Acesso em: 19/07/2010.
- OREILLY, TIM (2004). What Is Web 2.0. Disponível em: <oreilly.com/web2/archive/what-is-web-20.html>. Acesso em: 19/07/2010.
- OREILLY, TIM; BATTELLE, JOHN (2009). Web Squared: Web 2.0 Five Years On. Web 2.0 Summit 2009.
- PAUTASSO, CESARE; ZIMMERMANN, OLAF; LEYMAN, FRANK (2008). RESTful Web Services vs. “Big” Web Services: Making the Right Architectural Decision. World Wide Web Conference Committee (IW3C2).
- BLANCO, JOSE ARGUDO; UPTON, DAVID (2009). CodeIgniter 1.7. 1a Ed. Editora Packt Publishing, 2009.
- RICHARDSON, LEONARD; RUBY, SAM (2007). RESTful Web Services. 1a Ed. Editora O’Reilly Media, Inc., 2007.
- RFC2616 IETF (2010). HTTP, 2010. Em: <www.ietf.org/rfc/rfc2616.txt>. Acesso Em: 25/07/2010.