

# ERLANGMS: Uma Plataforma em Erlang/OTP para Modernização de Sistemas Legados através de uma Abordagem Orientada a Serviços na UnB

Everton Agilar<sup>1</sup>, Alysso Ribeiro<sup>1</sup>, Renato Ribeiro<sup>1</sup>, Eliene Vieira<sup>1</sup>,

<sup>1</sup>Universidade de Brasília – Campus Universitário Darcy Ribeiro  
Caixa Postal – 70910-90 – Brasília – DF – Brasil

{evertonagilar,rcarauta,alyssondsr,elienev}@unb.br

**Resumo.** Nos últimos anos, a modernização dos sistemas legados da Universidade de Brasília (UnB) tem sido prioridade para o CPD/UnB. A Arquitetura Orientada a Serviços (SOA) surge como uma maneira de solucionar este problema, disponibilizando uma abstração de alto nível entre as aplicações e a camada de negócio ou serviço. Este artigo aborda este tema e descreve alguns resultados obtidos com o uso da plataforma ERLANGMS desenvolvido pelo CPD/UnB sob uma abordagem orientada a serviços que compreende um processo de modernização e um barramento aderente ao estilo arquitetural Representational State Transfer (REST). Mais especificamente, são discutidos os principais resultados alcançados, as quais destacam-se um barramento de serviços orientado a contrato de serviços; um kit de desenvolvimento (SDK) para criação de serviços; um processo denominado SMSOC para guiar as atividades de migração de sistemas legados; e alguns serviços de apoio como um proxy LDAP para unificar o login de usuários e um serviço de autenticação/autorização OAuth2 para os serviços em REST.

## 1. Introdução

Os sistemas legados correspondem às aplicações que sustentam o funcionamento comercial de uma Instituição e consolidam a maior parte das informações corporativas [Bennett 1995, Bisbal et al. 1999]. Na Universidade de Brasília (UnB), há uma gama considerável de sistemas legados desenvolvido ao longo dos últimos 30 anos pelo CPD/UnB que consistem em um arcabouço de regras de negócios que são de vital importância para o pleno funcionamento da Instituição. Entretanto, com as sucessivas revisões nas regras de negócios para mantê-los alinhados com as necessidades e a obsolescência tecnológica desses sistemas, tornaram-se rígidos e inflexíveis, a ponto de serem de difícil manutenção e evolução.

De forma geral, os sistemas da UnB dividem-se em três áreas de negócio: área acadêmica, administrativa e de pessoal. A maioria desses sistemas foram construídos em diferentes linguagens de programação, arquiteturas e plataformas que não conversam entre si, a não ser, por meio do banco de dados. Durante muitos anos, a linguagem de programação VB foi a predominante. Os dois sistemas mais importantes escritos em VB são o *Sistema Acadêmico (SIGRA)* e o *Sistema de Pessoal (SIPES)*, sendo os demais sistemas escritos em VB.Net, C#, PHP, ASP e Java (a plataforma atual).

Neste cenário, as tradicionais práticas de manutenção deixam de atender às organizações, que buscam formas de reduzir os custos com a manutenção, maximizar a

integração entre os sistemas, torná-los mais flexíveis às mudanças de forma para prolongar sua vida útil e facilitar a evolução desses sistemas [Bennett 1995, Bisbal et al. 1999, Comella-Dorda et al. 2000].

Ao conduzir a modernização dos sistemas legados na UnB, optou-se por experimentar com a arquitetura orientada a serviços, particularmente seguindo o estilo arquitetural *REST*, já adotado em muitas Instituições devido ao aproveitamento da infraestrutura web existente e a facilidade para invocar serviços REST a partir de qualquer sistema [Fielding 2000, Kalin 2013]. Assim, foi criada a plataforma *ERLANGMS*, que compreende um barramento de serviços desenvolvido na linguagem *Erlang/OTP* para possibilitar a publicação de serviços; de um processo de modernização denominado *Software Modernization through Service Oriented Computing (SMSOC)*, para guiar os trabalhos de modernização e disponibilizar uma arquitetura de software padronizada para criação dos serviços.

Salienta-se que embora *SOA* seja um tema de crescente interesse por parte dos pesquisadores e da indústria, identificou-se a necessidade prévia de condução de um mapeamento sistemático para caracterizar a modernização de sistemas legados no contexto da manutenção de software [Agilar et al. ]. Desse modo, muitas decisões de design da plataforma proposta foram definidas a partir deste estudo prévio onde verificou-se que a maior parte das contribuições na literatura estão relacionados aos aspectos gerenciais da modernização de software (55,88% das publicações) e há poucos relatos de contribuições que descrevem (ou validem) técnicas ou ferramentas de modernização de software.

Este artigo apresenta a plataforma *ERLANGMS* e discute as experiências obtidas na implantação do barramento de serviços no CPD/UnB. Mais especificamente, são descritos os principais resultados alcançados, as quais destacam-se:

- Utilização do processo de modernização SMSOC
- Logon único para os sistemas da UnB
- Controle de acesso aos serviços
- 
- Desenvolvimento de um serviço de autenticação OAuth2, integrado ao Sistema de Controle de Acesso (SCA) da UnB, que permite que os serviços possam ser consumidos pelos clientes utilizando a mesma infraestrutura de segurança já existente.

## 2. Plataforma *ERLANGMS*

A plataforma *ERLANGMS* é constituído por um barramento de serviços (*Enterprise Service Bus*– ESB) multiplataforma, de um kit de desenvolvimento (SDK) para implementar os serviços na linguagem Java<sup>1</sup> e um processo de modernização denominado SMSOC. O barramento de serviços foi idealizado para servir de elo entre os sistemas da Universidade e a camada de serviço (tipicamente implementada usando a linguagem Java). De acordo com [Josuttis 2007], um barramento permite unificar o acesso aos serviços através de uma camada intermediadora entre componentes de software (denominados serviços) e as aplicações que consomem estes serviços. A implementação de um novo barramento (em vez da adoção de um barramento existente), possibilitou uma melhor compreensão do

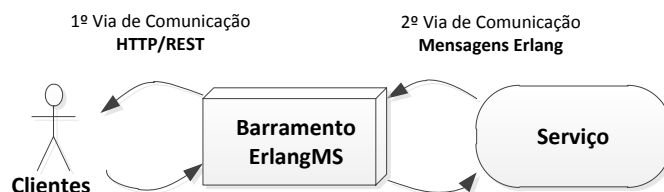
---

<sup>1</sup>Está em desenvolvimento o SDK .Net para implementação de serviços nas linguagens C# e VB.Net.

estilo arquitetural REST e o domínio de alguns elementos chave propostos no ErlangMS, como a estrutura de eventos e os recursos de tolerância a falha.

A arquitetura segue o conceito de *Service Oriented Computing* (SOC), um paradigma que promove a composição de serviços *em uma rede de serviços* fracamente acoplados, com o objetivo de criar processos de negócio dinâmicos e flexíveis através da interconexão de sistemas computacionais [Liebhart et al. 2010]. Dessa forma, o barramento suporta a mediação, roteamento, transformação de dados e a orquestração dos serviços. Para isso, adotou-se o estilo arquitetural REST e o formato JSON para o envio e recebimento das mensagens do cliente. Essa restrição de design teve o objetivo de facilitar a implementação do barramento e mantê-lo simples.

O esquema de comunicação da arquitetura ocorre por meio de duas vias distintas, como ilustra a Figura 1: Na primeira via, existe a comunicação do cliente para consumir algum serviço no barramento. Essa comunicação é via uma interface REST, razão pela qual o cliente (que pode ser qualquer sistema, independente da sua linguagem de programação ou plataforma) precisa suportar chamadas de serviços em REST. Na segunda via, tem a comunicação do barramento com o serviço, que está implementado em alguma linguagem de programação (Erlang, Java, etc.). Essa comunicação dá-se via sistema de mensageria disponível em Erlang que possibilita uma comunicação assíncrona com várias linguagens de programação de forma muito rápida por trafegar os dados no formato binário e com baixa latência na rede [Armstrong 2013].



**Figure 1. Esquema do roteamento das mensagens na plataforma ERLANGMS.**

Uma das vantagens que se observou na abordagem proposta foi permitir que tanto os sistemas novos quanto os sistemas legados possam coexistir, invocando os mesmos serviços e maximizando o compartilhamento das regras de negócio.

### 3. Resultados

#### 3.1. Utilização do processo de modernização SMSOC

A introdução de um processo surgiu devido a necessidade de documentar um processo de modernização para auxiliar os trabalhos de modernização. O processo SMSOC é aderente à arquitetura SOA e foi validado como resultado de um estudo de caso conduzido em uma disciplina de Pós-Graduação do Mestrado Acadêmico em Informática da UnB, através do qual foi modernizado o Sistema de Estudo Socio Economico (SAE) que faz a gestão do processo de avaliação socioeconômica dos estudantes.

O processo de modernização compreende também a documentação sobre a arquitetura e o design dos serviços implementados, denominado de kit de desenvolvimento

(SDK).

### **3.2. Logon único para os sistemas da UnB**

A UnB tem buscado desde 2013 uma forma de unificar o acesso dos usuários aos sistemas e a rede corporativa (Webmail e UnB Wireless), uma vez que existem muitos silos de dados de usuários que dificultam a centralização dessas informações. Para suprir esta demanda e subsidiar a concretização do acesso unificado ao ambiente da UnB bem como o compartilhamento de informações com várias as aplicações, até mesmo os sistemas não desenvolvidos pelo CPD/UnB, como é o caso do SEI e o Redmine, foi implementado um serviço proxy LDAP ou Lightweight Directory Access Protocol no barramento de serviços que permite que a base de usuários possa estar em um banco relacional onde os sistemas da UnB tem acesso direto mas permite que os demais sistemas desenvolvidos ou não pelo CPD e os sites institucionais possam autenticar os usuários e consultar informações desses usuários através de um protocolo padrão.

### **3.3. Controle de acesso aos serviços**

Com o desenvolvimento de sistemas com uma abordagem SOA, torna-se importante controlar o acesso e a autorização no acesso aos dados disponibilizados como serviço. Nesse sentido, foi desenvolvido suporte para autenticação OAuth2. Este protocolo ainda está em desenvolvimento e está sendo integrado ao sistema de controle de acesso da UnB.

ATÉ AQUI

## **4. Metodos**

Atualmente, os sistemas da Universidade de Brasília (UNB) são protegidos por meio de um sistema de controle de acesso (SCA). Esse sistema proporciona a criação de permissões através de usuários que tem perfis e estes tem transações. Um dos problemas enfrentados com essa abordagem é o fato de que as aplicações da UNB estão vinculadas a pessoa, ou seja, uma pessoa pode ter diversos usuários diferentes. Outro problema enfrentado é o fato desse sistema, por ter sido implementado de forma proprietária, não está em conformidade com os padrões do mercado no que se refere a autenticação e autorização, o que impossibilita que sistemas externos a universidade possam compartilhar dessas informações.

A implementação de um framework padrão de mercado se torna necessária para a resolução dos diversos problemas mencionados. Com isso foi escolhido o oauth 2 por ser um framework autamente utilizado pelas maiores empresas de informática e por ser hoje um padrão de mercado.

O oauth 2 é um framework que veio com a ideia de se padronizar a maneira como as aplicações cliente acessam conteúdo restrito. O oauth 2 se baseia em tokens de acesso. Ao se autenticar, uma aplicação cliente recebe do servidor de autorização um token de acesso. A aplicação cliente pode acessar os recursos restritos, basta apenas enviar o token recebido pelo servidor e este valida ou não o acesso aos dados requisitados.

Para a implantação do oauth 2 é necessário seguir uma série de passos preconizados pela rfc. Uma importante característica, que faz com que o oauth 2 seja amplamente utilizado é sua obrigatoriedade de uso junto com o HTTPS.

Diversos serviços já estão disponibilizados como: a declaração de aluno, o catálogo de aluno, os cursos dos alunos, dentre outros. Todos os serviços utilizam o barramento como comunicação entre as aplicações cliente e os dados armazenados no servidor.

Os serviços seguem um padrão, chamado de catálogos, que fornecem a url de acesso, qual a linguagem do serviço, onde está o código do serviço, seu método HTTP de acesso e a forma de autorização.

Segue abaixo um exemplo de um dos serviços já disponibilizados.

## 5. Resultados

### COLOCAR O OAUTH ACIMA

Para a integração do (SCA) com o protocolo oauth 2, o trabalho realizado teve como objetivo a criação de uma autenticação única e integrada para cada pessoa. Atualmente os perfis de acesso estão vinculados a um usuário, e vários usuários estão ligados a uma única pessoa. Para se ter um controle de acesso único e integrado é necessário associar todos os usuários a uma pessoa. Os seguintes passos foram feitos para padronização desse cenário.

Foi criado uma consulta que retornasse todos os perfis e usuários da pessoa. O resultado da consulta foi armazenado em um banco de dados interno do barramento para proporcionar maior velocidade na autenticação. Foi desenvolvida um sistema cliente de autenticação para o acesso único e integrado aos sistemas da UNB.

Os passos para a autenticação entre a aplicação cliente e o servidor de autorização seguem o seguinte fluxo. O usuário se autentica em uma aplicação cliente, que envia os dados de login e senha de acesso, por meio do SSL, ao receber o login e a senha o servidor autentica o usuário caso o login seja válido e retorna o token de acesso criptografado, que é armazenado pela aplicação cliente. Esse modelo permite que o usuário se autentique apenas uma vez e tenha acesso a todos os serviços que os perfis de acesso permitem.

## 6. References

### References

- Agilar, E., de Almeida, R. B., and Canedo, E. D. A systematic mapping study on legacy system modernization.
- Armstrong, J. (2013). *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf.
- Bennett, K. (1995). Legacy systems: coping with success. *Software, IEEE*, 12(1):19–23.
- Bisbal, J., Lawless, D., Wu, B., and Grimson, J. (1999). Legacy information systems: Issues and directions. *IEEE software*, 1(5):103–111.
- Comella-Dorda, S., Wallnau, K., Seacord, R., and Robert, J. (2000). A survey of legacy system modernization approaches. Technical Report CMU/SEI-2000-TN-003, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine.

Josuttis, N. M. (2007). *SOA in practice: the art of distributed system design*. " O'Reilly Media, Inc."

Kalin, M. (2013). *Java web services: up and running*. " O'Reilly Media, Inc."

Liebhart, D., Schmutz, G., and Welkenbach, P. (2010). Service-oriented architectureâan integration blueprint.