

# V SEMANA DO CONHECIMENTO

**CONSTRUINDO CONHECIMENTOS  
PARA A REDUÇÃO DAS DESIGUALDADES**

1 A 5 DE OUTUBRO DE 2018



**Marque a opção do tipo de trabalho que está inscrevendo:**

**Resumo**

**Relato de Caso**

## **PROJETO E IMPLEMENTAÇÃO DE UM SISTEMA WEB VOLTADO À ÁREA DE NUTRIÇÃO UTILIZANDO AS TECNOLOGIAS DOCTRINE E REST**

**AUTOR PRINCIPAL:** JOSÉ ÉRICO CAMERA SOARES

**CO-AUTORES:** JOSUÉ TOEBE E RAFAEL MARISCO BERTEI

**ORIENTADOR:** JOSUÉ TOEBE

**UNIVERSIDADE:** INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL  
RIOGRANDENSE - IFSUL

### **INTRODUÇÃO**

Em se tratando de desenvolvimento de sistemas de software, vivemos uma realidade em que busca-se cada vez mais a rapidez no projeto e implementação de produtos. Essa realidade se acentua com o surgimento de novos dispositivos e sistemas operacionais aos quais as aplicações devem se adaptar.

Como uma tentativa de minimizar essa dificuldade enfrentada por desenvolvedores, surgem novas ferramentas e frameworks. Essas opções de trabalho visam possibilitar maior agilidade em suas produções e aproximar os erros durante o decorrer do projeto a zero. Nesse sentido, dois métodos se destacam: O primeiro deles é o uso do modelo arquitetônico de web services REST (FIELDING, 2000) e o segundo é a ferramenta de mapeamento objeto-relacional Doctrine (DUNGLAS, 2013).

O objetivo deste trabalho foi a avaliação das tecnologias REST e Doctrine por meio da implementação de um software usual para os profissionais e estudantes da área de nutrição.

### **DESENVOLVIMENTO:**

O desenvolvimento do presente trabalho foi realizado em algumas etapas. Inicialmente buscou-se suporte na literatura para ampliar a compreensão das tecnologias e frameworks envolvidos. Em um segundo momento, realizou-se o

# V SEMANA DO CONHECIMENTO

**CONSTRUINDO CONHECIMENTOS  
PARA A REDUÇÃO DAS DESIGUALDADES**

1 A 5 DE OUTUBRO DE 2018



levantamento de requisitos e projeto do sistema. Finalmente, realizou-se a implementação e a avaliação dos resultados.

Durante a etapa de levantamento de requisitos e projeto utilizou-se a Linguagem de Modelagem unificada (UML). Assim, construiu-se os diagramas de casos de uso e de classes.

O protótipo de software foi desenvolvido seguindo as orientações de produção da arquitetura REST com o objetivo de, em situações futuras, caso exista a necessidade de integrá-lo com outras plataformas, assim fazê-lo e aproveitar todos os seus recursos disponibilizados pelos web services construídos. O sistema implementado com interface web dá apoio às necessidades básicas de um nutricionista, como, por exemplo, cadastro de clientes, controle de dietas, uma base de dados pré-pronta com grupos alimentares, entre outras.

O sistema desenvolvido poderá ficar disponível na web, assim vários profissionais poderão ter acessos a ele, cada um com seus clientes e controles. O sistema gerenciados de banco de dados utilizado foi Postgresql por se ajustar adequadamente com as tecnologias em avaliação.

Durante a implementação dos web services com REST percebeu-se uma possível falha na segurança, com a qual qualquer indivíduo que descubra URL do serviço solicitado poderia realizar uma requisição e obter os dados daquele serviço, como por exemplo, incluir, alterar ou até mesmo excluir dados de uma base de armazenamento. A solução encontrada para este problema foi a utilização de criptografia por intermédio do protocolo HTTPS e a implementação de um protocolo de segurança próprio.

Os passos do protocolo são apresentados a seguir:

- Usuário inicia comunicação com o servidor enviando usuário e senha;
- Servidor envia como resposta um token alfanumérico e armazena no cliente;
- A cada requisição do cliente, é enviado para o servidor o token juntamente com o endereço IP e a hora de acesso.
- Servidor verifica se o token do cliente é o mesmo gerado anteriormente e se o endereço IP coincide com o enviado ao primeiro acesso, se coincidirem, o usuário pode realizar a ação;
- Servidor então descarta token antigo e gera um novo, registrando ele na máquina do cliente;
- Por fim, ao sair do sistema, todos os token e dados como o IP são apagados, tanto do servidor quanto do cliente, reiniciando o ciclo em um novo acesso.

E no caso de dados incorretos, o servidor não autoriza a execução do comando e deleta todos os dados de acesso do cliente e no próprio servidor. Mesmo com esse método de segurança os dados ainda podem ser interceptados e adulterados, por isso a necessidade de uso do HTTPS.

## **CONSIDERAÇÕES FINAIS:**

O objetivo deste projeto foi alcançado. As duas tecnologias estudadas (Doctrine e REST) mostraram-se adequadas ao desenvolvimento. O software produzido encontra-



# V SEMANA DO CONHECIMENTO

**CONSTRUINDO CONHECIMENTOS  
PARA A REDUÇÃO DAS DESIGUALDADES**

1 A 5 DE OUTUBRO DE 2018



se em fase de implantação em um consultório e alguns detalhes ainda precisam ser corrigidos. Em se tratando do REST, resalta-se a necessidade da implementação de protocolos de segurança adicionais não disponibilizados pelos frameworks.

## REFERÊNCIAS

DUNGLAS, Kévin. Persistence in PHP with the Doctrine ORM. Birmingham, Reino Unido: Packt Publishing Ltd, 2013.

FIELDING, Roy T. Architectural Styles and the Design of Network-based Software Architectures. 2000. 162f. Dissertação (Doutorado em Filosofia) - Universidade da Califórnia - IRVINE, Califórnia, 2000.

**NÚMERO DA APROVAÇÃO CEP OU CEUA ( para trabalhos de pesquisa):**

## ANEXOS