

Utilização de um Leitor Dinâmico com autenticação JWT

Bruno de Amorim Campos, Lucas Melo Borges, Jean Carlos Gomes Aguiar

¹Instituto de Educação Superior de Brasília - IESB, Brasília - DF

{bruno.campos, lucas.borges, jean.aguiar}@iesb.edu.br

Resumo. *Este artigo traz uma autenticação envolvendo tokens JWT para validação de usuários, com o objetivo de permitir que realizem exercícios para leitura dinâmica. Nele é descrito as funcionalidades que foram utilizadas, descrevendo informações sobre como o token JWT foi devidamente utilizado.*

1. Introdução

Tendo em vista o aumento de tarefas no cotidiano, as pessoas tem obtido cada vez menos tempo para poderem se dedicar a uma leitura, seja por simples prazer, pela necessidade de estudos ou por conta de trabalho. Portanto, a busca por uma leitura mais rápida e eficiente tem crescido com a popularização da leitura dinâmica.

Órgãos como o Tribunal de Justiça do Estado do Rio de Janeiro por exemplo[TJRJ 2014], possuem cursos ensinando pessoas a se prepararem e melhorarem os seus desempenhos na leitura. Portanto é com esse objetivo em mente que este artigo apresenta um serviço onde os usuários poderão treinar suas capacidades de leitura dinâmica.

2. Objetivo Geral

O presente trabalho tem como objetivo propor uma solução *web* para auxiliar no treinamento e desenvolvimento de pessoas que possuem o interesse em obter uma leitura dinâmica

3. Objetivo Específico

Com base no interesse dos usuários de melhorar a sua capacidade de leitura e com isso treinar a mesma, foi então projetado as seguintes funcionalidades:

3.1. Upload de um arquivo PDF

O usuário será capaz de subir um arquivo em extensão .pdf para a aplicação, onde então ele irá treinar e ler o seu arquivo.

3.2. Controle de Velocidade

Uma vez com o arquivo na aplicação, o usuário será capaz de determinar a velocidade que deseja ler o seu texto, podendo assim controlando o quanto ele vai melhorando a sua capacidade.

4. Trabalhos Correlatos

Em busca de uma base para o projeto, encontrou-se alguns trabalhos correlatos, porém dois foram os que mais chamaram a atenção, sendo eles o Spreeder e o *Speed Reading App*.

4.1. Spreeder

Spreeder é uma aplicação web que possui como principal característica a capacidade de se colocar um texto inteiro e poder determinar a sua velocidade de leitura, sendo isso disponível desde já em sua versão gratuita.

Uma vez que se parte para a versão paga, ele permite que seja capaz extrair relatórios e fazer o *upload* de diferentes tipos de arquivos, facilitando o uso para quem o consome. Além disso, ele possui também uma aba dedicada a exercícios distintos para o aumento da eficácia da leitura.[Spreeder]

4.2. Speed Reading App

O *Speed Reading App* possui a vantagem de estar localizado no telefone celular do usuário, permitindo que o seu acesso seja de extrema facilidade. Porém, os usuários que não possuem o interesse em gastar com o *app*, estarão limitados a apenas um uso de texto diário, o que pode vir a desinteressar o uso do aplicativo.[Reading]

5. O Projeto

5.1. Linguagens

5.1.1. PHP

O PHP é uma linguagem de *script open source* de uso geral, sendo muito utilizada para o desenvolvimento *web* e podendo ser facilmente imbutida dentro de um HTML. [PHP]

Sendo a base da aplicação, o PHP possui a base controladora do projeto, onde ele faz o contato com o banco de dados e também realiza as verificações com o token necessário para poder efetuar o login de forma segura e correta.

5.1.2. Python

Sendo criado em 1991, o Python é uma linguagem de script tal qual o PHP. Ela possui a característica de ter uma tipagem dinâmica e que permite uma fluidez do seu desenvolvimento, além de possuir uma legibilidade alta, quando comparado a outras linguagens. [Org]

Seu uso no projeto foi para trabalhar com o upload e a interpretação dos arquivos PDF para o treino e leitura dos usuários. Ela foi escolhida pela sua capacidade e forte desempenho nesses quesitos.

5.2. JWT

O JWT ou *JSON Web Token*, é um método aberto utilizado para representar uma conexão de segurança entre dois pontos. Neste modelo, a informação ser verificada porque é assinada digitalmente. Os JWTs podem ser assinados com um segredo utilizando o algoritmo HMAC, ou com um par de chave pública e privada ao utilizar RSA ou ECDSA. [Auth0]

No caso do projeto, escolheu-se por utilizar uma criptografia RSA por já ser mais conhecida pelos membros do grupo e portanto, mais fácil de trabalhar com.

5.2.1. Estrutura

O JWT possui uma estrutura dividida em três partes que são separadas por '.', sendo elas:

- Cabeçalho
- Carga
- Assinatura

[Auth0]

Assim sendo, normalmente um JWT tipicamente possui o seguinte modelo:

xxxxxx.yyyyyy.zzzzzz

Neste modelo temos o cabeçalho que tipicamente consiste de duas partes, o tipo do *token* e o algoritmo de assinatura utilizado. Como na imagem a seguir:

Figura 1. Cabeçalho base JWT

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Fonte: [Auth0]

A segunda parte do *token* possui a carga enviada. É nela que é informado quais dados estão sendo repassados e os tipos deles, podendo pertencer a três: Registrado, público e privado.

Informações registradas são aquelas que já são padrão no JWT. Elas possuem apenas 3 caracteres para informar o seu dado e determinam as preferências do *token*. Enquanto isso, informações públicas são as que podem ser determinadas a vontade pelo usuário de JWT, mas com certo receio para não haver colisão com outros dados.

Por fim, os dados privados compartilham informações entre os receptores, de modo a não possuir dados registrados ou de conhecimento público. Normalmente são aqui que são armazenados as informações do usuário. [Auth0]

Figura 2. Exemplo de Carga JWT

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Fonte: [Auth0]

Por fim, o token JWT possui a assinatura utilizada que vai ser codificada e depois decodificada para ler as informações. Nessa parte, você utiliza de uma codificação das informações no cabeçalho e na carga, assim como o segredo utilizado. No site do JWT, tem-se o seguinte exemplo com uma codificação com o algoritmo HMAC SHA256.

Com isso, o token no final fica com o seguinte estilo:

Figura 3. Exemplo de Assinatura JWT

```
HMACSHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    secret)
```

Fonte: [Auth0]

Figura 4. Exemplo de Token JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4  
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.  
4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Fonte: [Auth0]

6. Aplicação

A aplicação foi desenvolvida de modo a auxiliar no treinamento de leitura dinâmica em um modelo de código aberto. Para isso, utilizou-se uma integração entre PHP e Python, que permitisse ao usuário que, ao subir um arquivo do tipo PDF, suas informações em texto fossem extraídas e armazenadas em um arquivo texto para melhor visualização e interpretação.

Figura 5. Script de conversão do arquivo de PDF para TXT

```
pdf = sys.argv[1]  
doc = fitz.open(pdf)  
pdf = pdf.replace('.pdf', '.txt')  
txt = open(pdf, "w")  
  
page = doc.loadPage(0)  
  
while(page.number != (len(doc) - 1)):  
    page = doc.loadPage(i)  
    text = page.getText('text')  
    text.encode('utf-8')  
    txt.write(text)  
    i += 1  
  
txt.close()
```

Fonte: autor

Após então realizada a conversão e transmissão do arquivo selecionado, o usuário encontra-se na tela de treinamento, onde o seu texto já foi devidamente selecionado e armazenado e está pronto para a execução. Com isso, basta o usuário determinar a velocidade de palavras por minuto que deseja treinar e apertar para iniciar.

Todo o fluxo dentro da aplicação, se dá por uma validação do Token JWT que foi gerado na hora do login. Páginas as quais não são necessárias, o token é ignorado. Porém,

Figura 6. Script de transmissão do novo arquivo para o leitor

```
arquivo = sys.argv[1]

with open(arquivo, 'r') as file:
    data = file.read()

palavra_processamento1 = data.replace(("-- ") or (" - ")), (" - ")
palavra_processamento2 = palavra_processamento1.replace("\n", " ")
palavras = data.split(" ")

print(palavras)
```

Fonte: autor

caso o usuário seja capaz de acessar páginas as quais sejam necessárias um token JWT, o mesmo não consegue desfrutar de suas informações caso o seu token seja inválido.

Essa validação se deu por meio de uma autenticação entre JavaScript e PHP, como mostram as imagens

Figura 7. Código que realiza a decodificação do token e verifica sua validade

```
$data = json_decode(file_get_contents("php://input"));

// get jwt
$jwt=isset($data->jwt) ? $data->jwt : "";

if($jwt){

    try {

        $decoded = JWT::decode($jwt, $key, array('HS256'));

        http_response_code(200);

        echo json_encode(array(
            "message" => "Acesso Concedido.",
            "data" => $decoded->data
        ));

    }catch (Exception $e){
        http_response_code(401);

        echo json_encode(array(
            "message" => "Access Negado.",
            "error" => $e->getMessage()
        ));

    }

}
```

Fonte: autor

Figura 8. Token JWT comumente enviado

```
$token = array(  
    "iat" => $issued_at,  
    "exp" => $expiration_time,  
    "iss" => $issuer,  
    "data" => array(  
        "id" => $user->id,  
        "firstname" => $user->firstname,  
        "lastname" => $user->lastname,  
        "email" => $user->email  
    )  
);
```

Fonte: autor

7. Conclusão

Com este projeto foi possível verificar a utilização de um Token JWT para a confiabilidade das informações transmitidas pelo usuário. Também foi possível validar a criação do leitor para realizar o treinamento de leituras dinâmicas.

Contudo, ao longo do projeto complicações foram encontradas com relação à validação do JWT, bem como a integração entre as linguagens.

Portanto, com o objetivo base em vista de criar um leitor dinâmico e de fácil utilização, pode-se concluir que o projeto obteve com êxito o seu objetivo.

Referências

Auth0. Introduction to JSON Web Tokens. <https://jwt.io/introduction>. [Online; acessado em 8-abril-2021].

Org, P. Python Help. <https://www.python.org/about/help/>. [Online; acessado em 13-abril-2021].

PHP. O que é o PHP? https://www.php.net/manual/pt_BR/intro-what-is.php. [Online; acessado em 17-abril-2021].

Reading, S. Speed Reading App: How to Read Faster. https://play.google.com/store/apps/details?id=com.speedreading.braintracers&hl=en_US&gl=US. [Online; acessado em 17-abril-2021].

Spreeder. Spreeder. <https://www.spreeder.com>. [Online; acessado em 17-abril-2021].

TJRJ (2014). Curso de Leitura Dinâmica. <http://www.tjrj.jus.br/documents/10136/19240/ementa-leitura-dinamica.pdf>. [Online; acessado em 17-abril-2021].