Universidade Federal da Bahia - UFBA
Instituto de Matemática - IM
Departamento de Ciência da Computação - DCC
Bacharelado em Ciência da Computação- BCC
Trabalho de Conclusão de Curso

Um encurtador de URLs para a Universidade Federal da Bahia

Pedro Augusto Vitor de Santana

Salvador - Bahia
Dezembro de 2019

Um encurtador de URLs para a UFBA

Pedro Augusto Vitor de Santana

Monografia apresentada como trabalho de conclusão de curso para o curso de Bacharelado em Ciência da Computação do Departamento de Ciência da Computação na Universidade Federal da Bahia.

Orientador: Prof. Dr. Rodrigo Rocha Gomes e Souza.

Salvador - Bahia
Dezembro de 2019

Um encurtador de URLs para a Universidade Federal da Bahia

Rodrigo Rocha Gomes e Souza

Monografia apresentada como trabalho de conclusão de curso para o curso de Bacharelado em Ciência da Computação do Departamento de Ciência da Computação na Universidade Federal da Bahia.

Banca Examinadora:

Prof. Dr. Rodrigo Rocha Gomes e Souza (Orientador)
UFBA

Prof. Dr. Leobino Nascimento Sampaio
UFBA

Elmo do Vencimento Baraúna
Unijorge

À minha família

Agradecimentos

Primeiro agradeço à minha mãe, que desde o dia em que eu nasci, doa-se de forma surpreendente e me ajudou em cada passo dessa caminhada. À minha vó Bah, que apesar de eu lamentar não conseguir concluir isso em vida para te mostrar, que você esteja vendo isso de um ótimo lugar. Agradeço à meus 3 irmãos. Agradeço também à minha namorada Ana Luísa que sempre esteve ao meu lado me apoiando ao longo de toda a minha trajetória. À Mirelle e Davi que foram os seres humanos mais acolhedores que tive o prazer de conviver. Por fim agradeço ao meu orientador Rodrigo Rocha não só por sua excelência técnica, mas também por ser uma pessoa exemplar em todos os aspectos.

"Pânico de nada, é tudo estrada"
Don L

Resumo

O encurtamento de URLs é uma técnica onde o tamanho da URL é substancialmente reduzido. Sua utilização tem diversas razões, a maioria ligada a necessidade de se divulgar endereços por meios que limitam o tamanho de mensagens. Outra utilização comum é para facilitar a digitação de determinadas URLs que estão impressas em jornais, revistas, artigos científicos e etc.

Junto com a redução do tamanho, a URL curta possui o domínio diferente da original e isso pode ser utilizado para ocultar o domínio da mesma. Este trabalho tem como objetivo fornecer a Universidade Federal da Bahia um serviço de encurtamento de URL que possa ser utilizado pelos membros da rede acadêmica, buscando oferecer o máximo segurança para que não haja a utilização por usuários com objetivos maliciosos.
Palavras-chave: <Encurtamento de URL>, <Links>, <AUFBA>, <URL>.

Abstract

URL shortening is a technique where URL length is substantially reduced. Its use has several reasons, mostly linked to the need to disclose addresses by means that limit the size of messages. Another common use is to make it easy to type in certain URLs that are printed in newspapers, magazines, scientific articles, and so on.

Along with reducing the size, the short URL has a different domain from the original one and this can be used to hide the domain from it. This paper aims to provide the Federal University of Bahia with a URL shortening service that can be used by members of the academic network, seeking to offer maximum security against the use by users with malicious purposes. Keywords: <URL>, <Shortener>.

Sumário

Lista de Figuras

Lista de Tabelas

1. Introdução

Com a necessidade de se reduzir o tamanho de URLs por vários motivos, sendo alguns deles a divulgação em meios que limitam a quantidade de caracteres ou até mesmo a digitação dessas URLs para ter acesso ao conteúdo, surge os encurtadores de URLs, que conseguem reduzir substancialmente o tamanho da URL consegue ainda assim chegar ao mesmo objetivo.

Neste trabalho foi feito a implementação de um serviço com as características citadas acima e que pode ser usada por todos os membros da rede acadêmica da UFBA, pois durante a sua implementação foram considerados ao máximo os pontos julgados necessários, essas considerações são amplamente discutidas ao decorrer do trabalho. O objetivo das discussões é definir métodos que busquem extinguir a utilização da ferramenta para fins que infrinjam qualquer norma da instituição ou que exponha seus utilizadores à riscos de qualquer natureza.

Os passos que anteciparam o desenvolvimento foram pesquisas por trabalhos relacionados, onde surgem muitas questões que inicialmente não eram visualizadas como parte do problema, é o caso do Linkrot ou apodrecimento de URLs, onde os recursos apontados por URLs tendem a mudar, serem movidos ou removidos ao longo do tempo. Outras questões eram mais óbvias e foram apontadas antes das pesquisas.

Foi definido que a aplicação seriam mantida na infraestrutura do STI-UFBA, isso possibilita a aplicação uma expectativa de vida maior e também induz a posse da aplicação para a Universidade. A solicitação de um subdomínio UFBA para a aplicação foi feita e atendida, com isso a aplicação ficou localizada no endereço: `https://a.ufba.br`. Junto com os requisitos exigidos pela STI e os estudos feitos são definidos os requisitos da aplicação. As etapas posteriores foram tomadas de decisões, pois haviam possibilidades diversas. Construir uma aplicação do zero ou utilizar um projeto de código aberto que se encaixasse nos requisitos. Após analisar e escolher utilizar um projeto existente surgiu a decisão de qual dos projetos disponíveis se encaixava melhor na proposta e foi definida a utilização de um projeto bem consolidado, com uma documentação boa e de fácil implementação de melhorias.

O projeto escolhido precisava de implementações adicionais para que pudesse cumprir com os requisitos estabelecidos, como permitir autenticação usando as credenciais da UFBA, formas de bloqueio de URLs, definição de comportamentos baseados na URL pertencer ou não ao domínio UFBA entre outras necessidades levantadas durante o an-

damento trabalho.

Ao final do projeto foi possível observar uma aplicação totalmente funcional, que atendeu à todos os requisitos levantados, possibilitando também a utilização do trabalho não apenas para a melhoria da própria aplicação, mas também a utilização da mesma como uma ferramenta base de apoio à outras soluções.

2. Fundamentação Teórica

Neste capítulo serão apresentados os conceitos necessários para o entendimento completo do trabalho. A ideia principal é fornecer todas as informações que foram julgadas importantes para a compreensão da solução.

2.1 Localizador Uniforme de Recursos

O Localizador Uniforme de Recursos (URL) se refere ao endereço de rede no qual se encontra algum recurso informático, como por exemplo, um arquivo de computador ou um dispositivo periférico (impressora, equipamento multifuncional, unidade de rede etc.). Essa rede pode ser a Internet, uma rede corporativa (como uma intranet) etc.

Em geral as URLs tem a seguinte estruturas[1]:

{Esquema} : {Parte específica do esquema}

As opções para esquema de acordo com o RFC 1738 são:

- FTP
- FILE
- HTTP
- GOPHER
- MAILTO
- NEWS
- NNTP
- TELNET
- WAIS
- PROSPERO

Cada esquema tem sua própria parte específica. Nós iremos definir aqui apenas o Hypertext Transfer Protocol (HTTP), pois será o único abordado com profundidade. O esquema segue a seguinte estrutura.

{http} :// {domínio} : {porta} / {caminho} ? {busca}

- **HTTP** - HTTP é um protocolo que permite a obtenção de recursos, tais como documentos HTML. É a base de qualquer troca de dados na Web e um protocolo cliente-servidor, o que significa que as requisições são iniciadas pelo destinatário, geralmente um navegador da Web. Um documento completo é reconstruído a partir dos diferentes sub-documentos obtidos, como por exemplo, texto, descrição do layout, imagens, vídeos, scripts e muito mais.

- **Domínio** - O nome de domínio totalmente qualificado de um host de rede ou seu endereço IP como um conjunto de quatro grupos de dígitos decimais separados por ".". O nome domínio é uma sequência de rótulos de domínio separados por ".". Cada rótulo de domínio começa e termina com um caractere alfanumérico e possivelmente também contendo caracteres -". O rótulo do domínio mais à direita nunca começará com um dígito, que distingue sintaticamente todos os nomes de domínio dos endereços IP.

- **Porta** - O número da porta à qual se conectar. A maioria dos esquemas designa protocolos que possuem um número de porta padrão. Outro número de porta pode opcionalmente ser fornecido, em decimal, separado do domínio por dois pontos. Se a porta for omitida, os dois pontos também serão. No caso do HTTP a porta padrão é a 80.

- **Caminho** - O caminho especifica o local (geralmente num sistema de arquivos) onde se encontra o recurso, dentro do servidor.

- **Busca** - Também chamada de "query string" é um conjunto de um ou mais pares "pergunta-resposta" ou "parâmetro-argumento" (como por exemplo nome=fulano, em que nome pode ser, uma variável, e fulano é o valor atribuído a nome). É uma string enviada ao servidor para que seja possível filtrar ou mesmo criar o recurso.

2.2 Encurtadores de URL

O encurtamento de URL é uma técnica na Rede Mundial de Computadores na qual uma URL pode ter um comprimento substancialmente mais curto e ainda direcionar para a página necessária[2]. Isso é obtido usando um redirecionamento HTTP em um nome de domínio curto, vinculado à página da web que possui uma URL longo. Isso é especialmente conveniente para tecnologias de mensagens como Twitter e Identi.ca, que limitam severamente o número de caracteres que podem ser usados em uma mensagem. URLs curtos permitem que endereços da web longos sejam mencionados em um tweet.

Os encurtadores mais populares utilizam a seguinte estrutura [3]:

{protocolo} :// {domínio} / {identificador}

- **Protocolo** - O protocolo que pode ser http ou https que é uma implementação do primeiro sobre uma camada adicional de segurança que permite que os dados sejam transmitidos por meio de uma conexão criptografada e que se verifica a autenticidade do servidor e do cliente através de certificados digitais e utiliza o protocolo SSL/TLS.
- **Domínio** - O domínio é o endereço da aplicação na internet.
- **Identificador** - O identificador é a palavra chave, que é identificada unicamente e a partir dela a aplicação irá retornar o endereço que será redirecionada a página, caso esse endereço esteja na base de dados.

A definição do identificador é feita de duas formas, sendo a primeira gerada pela própria aplicação. O sistema persiste a URL longa na base de dados e retorna esse identificador informando ao usuário que ao acessar o endereço composto por esse identificador ele será redirecionado para o endereço que ele informou. Na segunda forma, a palavra chave é definida pelo usuário, preenchendo além da URL destino, um conjunto de caracteres que será utilizado para acessar essa URL destino.

A previsibilidade de URLs é um problema a ser considerado, pois usuários do serviço costumam não entender que a partir do momento que uma URL é encurtada, ela pode ser acessada sem restrição por qualquer pessoa que tenha acesso ao endereço. Um estudo [3] compilado em maio de 2011 por pesquisadores da Universidade de Aachen, na Alemanha, coletou informações de vários serviços de encurtadores de URLs mostrou que usuários utilizam os serviços para encurtar URLs sensíveis, foram encontrados 71 documentos publicamente acessíveis hospedados pelo Google Documents, fotos particulares, um documento de seminário, um relatório do tesoureiro de uma empresa, dois curriculum vitae e a lista de nomes, endereços e números de telefone de um jardim de infância. Com isso tende-se a dificultar a previsibilidade dos identificadores a fim de evitar que terceiros vasculhem as URLs curtas buscando encontrar alguma informação sensível.

Um outro uso do encurtamento de URL é disfarçar o endereço real. Embora isso possa ser desejado por motivos pessoais ou comerciais legítimos, ele permite abusos e por esse motivo, alguns provedores de serviços de encurtamento de URL foram incluídos em listas negras de spam, devido ao uso de seus serviços de redirecionamento por sites com o intuito de burlar essas listas. Alguns sites impedem que URLs curtos e redirecionados sejam postados e isso fez com que algumas instituições implementassem seu próprio serviço de encurtamento.

O Twitter enxergou a necessidade de se criar seu próprio encurtador e em 2011 lançou o serviço t.co [4]. Segundo a pĺataforma, a razão para o desenvolvimento do seu serviço são as seguintes:

1. Os links encurtados permitem compartilhar URLs longos em um Tweet sem comprometer o número máximo de caracteres da mensagem.

2. O serviço de links avalia informações como o número de vezes que um link foi clicado, o que é um sinal de qualidade importante para determinar a relevância e o interesse apresentados por cada Tweet em relação a Tweets similares.

3. O encurtador de links protege os usuários contra sites maliciosos que propagam malware, ataques de phishing e outras atividades prejudiciais. Os links convertidos pelo serviço de links do Twitter são verificados com base em uma lista de sites potencialmente perigosos. Os usuários são avisados com uma mensagem de erro quando clicam em URLs potencialmente perigosos.

2.3 Blacklists

Uma blacklist ou lista negra é uma abordagem básica de controle de acesso que permite acesso de todos os elementos, exceto aqueles contidos na lista negra. Esses itens da lista têm acesso negado.

As listas negras podem ser aplicadas em vários pontos de uma arquitetura de segurança, como host, proxy da web, servidores DNS, servidor de e-mail, firewall, servidores de diretório ou gateways de autenticação de aplicativos. O tipo de elemento bloqueado é definido baseado no contexto. Quando estamos falando de uma blacklist para e-mails, o que pode ser filtrado são endereços de e-mail do remetente, ou um conjunto de palavra no corpo do e-mail, já ao falarmos em um contexto de firewall, o que serão filtrados são IPs, domínios, arquivos, usuários. As blacklists abordadas aqui terão como objetivo filtrar URLs.

Em 2008 o Centro de Atendimento a Incidentes de Segurança (CAIS) área de segurança da informação da Rede Nacional de Ensino e Pesquisa (RNP) criou o Catálogo de Fraudes da RNP[5] que é um serviço de identificação e catalogação de e-mails fraudulentos. O Catálogo de Fraudes da RNP é mantido pelo CAIS numa parceria com um grupo de pesquisadores e técnicos da UFBA e do Ponto de Presença da RNP na Bahia (PoP-BA/RNP) e permite que qualquer pessoa faça a validação de um e-mail suspeito.

Ao analisar as fraudes que circulam via e-mail no Brasil e que são reportadas ao Catálogo de Fraudes da RNP, o grupo de pesquisadores e técnicos que atuam na manutenção do catálogo notou a presença frequente de URLs maliciosas nos e-mails e que as ferramentas de contenção dessas URLs apresentam baixa taxa de detecção no contexto de fraudes analisadas. Esses fatos culminaram na proposta de criação do Catálogo de URLs Maliciosas (CaUMa) cujo objetivo é prover à comunidade um serviço adicional para

combate aos sites fraudulentos, a partir do seu bloqueio em navegadores web e clientes de e-mail.

O CaUMa permite tanto a consulta do usuário através de uma interface Web, quanto através de uma requisição HTTP para sua API, construída visando a utilização do catálogo por outras aplicações.

2.4 Linkrot

Linkrot é o fenômeno de URLs tendendo ao longo do tempo a deixar de apontar para seu arquivo, página da web ou servidor originalmente direcionado, devido a esse recurso ser realocado ou ficar permanentemente indisponível [6]. Uma URL que não aponta mais para o destino, geralmente chamado de link quebrado ou morto.

A página da web de destino pode ser removida. O servidor que hospeda a página de destino pode falhar, ser removido do serviço ou realocado para um novo nome de domínio. O registro de um nome de domínio pode expirar ou ser transferido para outra pessoa, algumas falhas retornaram o código HTTP 404 [7] enquanto outras podem simplesmente direcionar pra um recurso diferente do original. Algumas outras causas são:

- A reestruturação de sites que causa alterações nos URLs
- Realocação de conteúdo anteriormente gratuito para um ambiente pago
- Uma mudança na arquitetura do servidor que resulta em código
- Conteúdo dinâmico da página, como resultados de pesquisa que mudam de design
- A presença de informações específicas do usuário, como um nome de login no link
- Bloqueio deliberado por blacklists

Um estudo de 2014 da escola de direito de Harvard analisa as implicações legais da quebra de URLs na Internet [8] e encontra razões para alarme.

Os autores, Jonathan Zittrain, Kendra Albert e Lawrence Lessig, determinaram que aproximadamente 50% dos URLs nas opiniões da Suprema Corte dos Estados Unidos da América não vinculam mais as informações originais. Eles também descobriram que em uma seleção de periódicos jurídicos publicados entre 1999 e 2011, mais de 70% dos links não funcionavam mais como pretendido. Os estudiosos escrevem:

> Conforme os sites evoluem, nem todos os terceiros terão interesse suficiente em preservar os links que fornecem compatibilidade com versões anteriores para aqueles que confiaram nesses links. O autor da fonte citada pode decidir

> que o argumento está incorreto e retirá-lo. O proprietário do site pode decidir abandonar um modo de organização de material para outro. Ou a organização que fornece o material de origem pode alterar suas visões atualizar a fonte original para refletir suas visões em evolução. Em cada caso, o documento que cita é vulnerável a notas de rodapé que não apoiam mais suas reivindicações. Essa vulnerabilidade ameaça a integridade da bolsa de estudos resultante[8].

Como encurtadores de URLs agem como uma camada intermediária entre a URL longa e a URL curta, em caso do serviço parar de funcionar por qualquer motivo, todas as URLs curtas pararam de redirecionar para a URL longa e isso contribui bastante com o problema de linkrot.

2.5 Autenticação UFBA

Autenticação UFBA é uma aplicação desenvolvida pela STI que permite a autenticação de integrantes da UFBA em aplicações diversas, utilizando suas credenciais da UFBA.

A ferramenta é escrita utilizando o protocolo Central Authentication Service (CAS). CAS é um protocolo de logon único para a web. Seu objetivo é permitir que um usuário acesse vários aplicativos enquanto fornece suas credenciais (como ID do usuário e senha) apenas uma vez.

Com o CAS também é permitido que aplicativos da web autentiquem usuários sem obter acesso às credenciais de segurança de um usuário, como uma senha. O nome CAS também se refere a um pacote de software que implementa este protocolo.

3. Solução

A solução foi pensada tendo dois princípios em mente. O primeiro seria que ao se tratar de uma aplicação que ao ser utilizada serve como uma ponte de acesso a dados, ela precisa ser permanente, pois se em algum momento ela parar de funcionar toda a informação que foi compartilhada utilizando-a será perdida. Uma forma de conseguir que a aplicação esteja sempre disponível foi passar a responsabilidade da sua hospedagem para a Superintendência de Tecnologia da Informação da UFBA (STI), pois ela fornece suporte desde o domínio, até a hospedagem da aplicação e o banco de dados e isso nos leva ao segundo princípio.

Ao utilizarmos os serviços da STI, a aplicação pertencerá ao subdomínio da UFBA e com isso temos a responsabilidade de impedir que ela seja utilizada para fins que não condizem com as normas da instituição.

3.1 Prevenções

Fornecendo ao usuário a possibilidade de criar uma URL com uma quantidade reduzida de caracteres tem-se como principal objetivo facilitar na divulgação do endereço, desde o uso em redes sociais que limitam os caracteres por mensagem, até cenários onde é necessário digitar esse endereço para acessá-lo. Fontes encontradas em trabalhos científicos, revistas e jornais impressos são alguns dos exemplos.

Ao permitir que um endereço não pertencente ao domínio UFBA seja redirecionado a partir de um outro endereço que a ele pertence surgem alguns problemas de segurança.

3.1.1 Abusos

O encurtador pode vir a ser utilizado por pessoas má intencionadas como uma forma de camuflagem para atividades maliciosas ou criminosas.

Temos listas negras de endereços que fazem o trabalho de catalogar endereços maliciosos e bloqueá-los. Ao permitir que um endereço cadastrado nessas listas seja encurtado estamos facilitado esse tipo de ação, pois é esperado que nosso domínio não pertença a nenhum catálogo e ainda assim um endereço encurtado pode ser levado a sites catalogados. [3]

3.1.2 Bloqueios e Banimentos

Encurtadores de URLs são uma forma de burlar blacklists[9]. Utilizando o intermédio do serviço que normalmente não está presente nesses catálogos, é possível divulgar endereços utilizando a URL encurtada, e assim não ser identificado. Em situações onde o site em que está localizado o encurtador é incluído em blacklists, a depender da abordagem utilizada por ela, todo o conteúdo dele é impedido de ser acessado.

O Wikipédia tem incluso em seu catálogo de spam uma categoria só para encurtadores de URLs [10], isso implica em não permitir nenhuma URL desses serviços na plataforma. Dessa forma é necessário mecanismos de controle para que a aplicação não seja utilizada de forma que leve-a ser listada em alguma blacklist.

3.1.3 Privacidade e Segurança

Ao acessar um endereço encurtado estamos disponibilizando uma série de informações ao serviço. O simples ato de criar a URL encurtada pode permitir que motores de buscas indexem esse conteúdo a partir do hiper link de forma a dar a acesso de várias pessoas a esse link.

O usuário pode não ter conhecimento que ao utilizar um serviço de encurtamento, o endereço utilizado deixa de ser privado e pode ser encontrado por qualquer pessoa. A depender dos mecanismos implementados pela aplicação, isso pode ser mais fácil ou difícil.

3.1.4 Previsibilidade de identificadores gerados automaticamente

Ao gerar um identificador sequencial, é muito fácil definir uma faixa de identificadores que foi utilizado por outros usuários. Como visto na seção 2.2, é comum a utilização do serviço para o encurtamento de URLs que contém dados sensíveis. Assim concluímos a necessidade de se restringir ao máximo a previsibilidade de URLs curtas, para dificultar a tentativa por força bruta com o objetivo de prever a identificação de endereços encurtados a fim de obter esse dados de outros usuários que não deveriam estar expostos.

3.2 Implementação

Depois de listado todos os requisitos da solução, o processo de implementação deve levar em consideração todas as necessidades levantadas. Levando em consideração que a solução estará hospedada nos servidos do STI, foi necessário listar restrições que a aplicação precisava respeitar:

1. Linguagem PHP - Versão 7.2.

2. Sistema de gerenciamento de banco de dados (SGBD) MySQL.

Partindo dessas restrições foi necessário escolher entre implementar toda a aplicação ou utilizar um projeto já existente e fazer as modificações que se julgar necessária.

3.2.1 Implementar totalmente

O funcionamento de um encurtador de URLs consiste basicamente em quando o usuário fornecer a URL que ele deseja encurtar. A depender da decisão de projeto, o identificador seção 2.2 pode ser gerado automaticamente ou escolhido na hora da criação. Após validação, é salva numa base de dados a URL a ser encurtada e o identificador correspondente. Na hora de acessar uma URL curta, procura-se na base de dados qual o endereço correspondente ao identificador que foi inserido. Depois disso, redireciona o usuário.

Ao implementar toda a aplicação partindo do zero, seria necessário estudar as necessidades de um encurtador, definir arquitetura da aplicação, padrão de projeto, usar ou não frameworks e modelagem entidade-relacionamento. Ao fazer essa escolha, haveria total liberdade no desenvolvimento da aplicação, possibilitando gerir o foco para questões que só surgiram no contexto da nossa aplicação, como, por exemplo, a autenticação por meio da API disponibilizada pela UFBA [11].

3.2.2 Utilizar projeto existente

O serviço de encurtamento de URLs é bem popular [12], com isso é possível encontrar uma série de projetos que já implementam a solução. Após filtrar projetos que são escritos em PHP e que permitem utilização do MySQL, como SGBD, permaneceram apena dois: o Polr e o Yourls.

3.2.3 Polr vs Yourls

A princípio a informação era que a versão do servidor que ficaria hospedada a aplicação, fornecia suporte apenas para PHP 5.3.3. Sendo assim, a opção mais viável era o Yourls, pois o Polr necessita do PHP com versão igual ou superior a 5.5.9 impossibilitando sua implementação.

Após o processo de solicitação de hospedagem e domínio junto a STI, foi visto que a versão do PHP presente no servidor era 7.3.10 e com isso foi necessário comparar os dois projetos e ver qual se adequava mais as necessidades.

3.3 Yourls

O Yourls "é um pequeno conjunto de scripts PHP que permitirá que você execute seu próprio serviço de encurtamento de URL (à la TinyURL ou Bitly). A execução do seu próprio encurtador de URL é divertida, nerd e útil: você possui seus dados e não depende de serviços de terceiros. Também é uma ótima maneira de adicionar marca aos seus URLs curtos, em vez de usar o mesmo encurtador de URL público que todos usam" [13].

3.3.1 Requisitos

Na página oficial do projeto, é possível encontrar várias informações sobre ele, inclusive esses são os requisitos informados para a versão que estamos utilizando:

1. Versão 1.7.3
2. PHP 5.3 ou superior
3. MySQL 5 ou superior
4. Servidor Apache, Nginx ou Cherokee

3.3.2 Recursos

Esses são os recursos listados na página com isso foi possível identificar as primeiras comparações com o Polr

- Grátis e código aberto
- Privado ou Público
- Identificadores sequenciais ou personalizados
- Bookmarklets úteis para encurtar e compartilhar links com facilidade
- Históricos de cliques, rastreamento de referências, localização geográfica dos visitantes
- Interface dinâmica utilizando Ajax.
- Arquitetura de plug-ins para implementar facilmente novos recursos
- API para desenvolvedores
- Suporte jsonp completo

- Instalação amigável
- Arquivos com exemplos

3.4 Polr

Polr "é um encurtador de links de código aberto rápido, moderno e de código aberto. Ele permite que você hospede seu próprio encurtador de URL, identifique seus URLs e obtenha controle sobre seus dados. Também possui licença GPLv2+" [14].

3.4.1 Requisitos

A primeira release do Polr é de 2015, enquanto o Yourls tem sua primeira versão lançada em 2009. É observado que o Polr é mais moderno, mas tem muito mais requisitos, como mostrado:

1. Apache, nginx, IIS, or lighttpd (Preferencialmente Apache)
2. PHP 5.5.9 ou superior
3. MariaDB or MySQL 5.5 ou superior
4. composer
5. Requisitos PHP:
   (a) OpenSSL PHP Extension
   (b) PDO PHP Extension
   (c) PDO MySQL Driver (php5-mysql on Debian & Ubuntu, php5x-pdo_mysql on FreeBSD)
   (d) Mbstring PHP Extension
   (e) Tokenizer PHP Extension
   (f) JSON PHP Extension
   (g) PHP curl extension

3.4.2 Recursos

Por ser mais novo o Polr utiliza tecnologias mais recentes e também é possível utilizar uma demonstração do projeto[15]. Tem os seguintes recursos informados:

- Escrito em PHP e alimentado pelo microframework Lumen

- Usa template Blade
- Arquitetura MVC limpa
- Usa a ORM Eloquent (MySQL, PostgreSQL ou SQLite)
- Polr é intrépido, fácil de usar
- Leia a documentação de instalação e faça funcionar rapidamente
- Interface moderna e simples para gerenciar seus links e controlar sua instância
- Execute em seu próprio domínio para ajustar sua marca à perfeição
- Use Polr fora da caixa ou bifurque o código para ajustá-lo às suas necessidades
- Personalize as permissões de encurtamento, redirecionamentos ou mesmo o tema da sua instância
- API REST semântica para integração com outros serviços
  - Atribua novas chaves de API aos usuários automaticamente ou gere-as manualmente
  - Crie novos links ou procure links existentes
  - Para mais informações, consulte a documentação da API
- Instância do Polr Demo: demo.polr.me
  - Nome de usuário: demo-admin ou demo-user
  - Senha: demo-admin ou demo-user (a mesma do nome de usuário)
  - Certos recursos, como exclusão do usuário e alteração de senha, são desativados na instância de demonstração
  - Use a instância demo para testar a API, a interface ou o painel de administração

3.5 Comparativo

O Polr foi escolhido inicialmente pela estrutura do código, principalmente por utilizar a arquitetura MVC e facilitar muito as alterações que seriam necessárias implementar.

	Yourls	Polr
PHP	>= 5.3	>= 5.5.9
MySQL	>= 5.0	>= 5.5
API	✓	✓
Boa documentação	✓	✓
Plugin	✓	✓
Framework	x	✓
ORM	x	✓
QRCode	x	✓
Funcionou na STI	✓	x

Tabela 3.1: Comparativo entre Yourls e Polr.

Depois das primeiras alterações serem feitas, ao tentar fazer a publicação no servidor de produção, houve bastante dificuldade para fazer funcionar, pois a estratégia da STI é a utilização de FTP e eram necessárias alterações no ambiente que exigiam mais que apenas a transferência dos arquivos que compõem a aplicação. Com isso, foi identificado que o YOURLS necessitava de menos recursos, tornando possível fazer a publicação apenas transferindo os arquivos do projeto para o servidor. Sendo assim, houve a migração para o mesmo.

3.6 Restringindo acesso

Como discutido na seção 3.1 não deve ser permitido a utilização da aplicação para fins que não condizem com as normas da instituição. Desse modo, deve ser negada ao máximo qualquer tentativa de fazê-la.

As URLs a serem encurtadas foram divididas em dois grupos. O primeiro diz respeito a qualquer uma que pertença ao domínio UFBA e o segundo são todas as outras que não pertencem. Dessa forma, podemos fornecer a liberdade da utilização da aplicação para as URLs pertencentes ao primeiro grupo, pois foi assumido que se já estivesse dentro do domínio, não estaria sendo atribuído privilégio algum encurtando-a que já não estivesse a ela atribuída em sua forma longa.

O segundo grupo trouxe a tona problemáticas que precisavam ser tratadas. Como permitir que a aplicação pudesse ser utilizada para esse grupo, sem que pudesse ser utilizada de forma indiscriminada?

3.7 Alterações

Com as restrições adicionadas, ainda havia o problema de previsibilidade das URLs, além da necessidade de permitir que sejam bloqueados determinados links caso seja identificado que esse link desrespeita alguma norma.

Foi necessário implementar uma forma de autenticar o usuário utilizando o Autenticação UFBA, assim com a utilização de um serviço onde podemos vincular um usuário à uma conta já existente sem que seja necessário ao usuário nos fornecer nenhum dado diretamente. Foi possível resolver o problema das URLs que não pertencem ao domínio UFBA, apenas exigindo que o usuário esteja autenticado antes de encurtá-las e vinculando cada URL criada ao usuário que estava logado no momento da criação. Isso nos garante resolver boa parte dos problemas, mas não todos.

Foi utilizada a biblioteca phpCAS, que permite fazer a comunicação com o servidor da Autenticação UFBA. A biblioteca permite verificar se o usuário está autenticado e caso esteja, retorna seu nome de usuário.

Para ser possível auditar as URLs maliciosas, foi implementado um vínculo entre as URLs curtas e os usuários logados no momento da criação, dessa forma temos o registro de qual usuário criou, Nesse caso ao identificar uma URL maliciosa pode buscar todas as URLs do mesmo criador e desativá-las se necessário.

3.7.1 Geração de identificadores aleatórios

Por conta da utilização dos encurtadores em endereços que levam a informações sensíveis, há a necessidade de se adicionar uma camada extra de segurança, pois a geração de identificadores fornecidas pelo YOURLS é na forma sequencial, quando não personalizada. Utilizando a estrutura vista na sessão 2.2. Dessa maneira sabendo um identificador é facilmente visualizado qual o próximo ou o anterior, pois a geração desses identificadores segue a mesma lógica do sistema de numeração hexa-trigesimal[16] onde a ordem é da seguinte forma:

Tabela 3.2: Base 36

Decimal	**Hexatrigesimal**
0	0
1	1
2	2
3	3
Continua na próxima página	

Tabela 3.2 – continuação da página anterior	
Decimal	**Hexatrigesimal**
4	4
5	5
6	6
7	7
8	8
9	9
10	a
11	b
12	c
13	d
14	e
15	f
16	g
17	h
18	i
19	j
20	k
21	l
22	m
23	n
24	o
25	p
26	q
27	r
28	s
29	t
30	u
31	v
32	w
33	x
34	y
35	z
36	10
Continua na próxima página	

Tabela 3.2 – continuação da página anterior	
Decimal	**Hexatrigesimal**
37	11
38	12

É fornecido também a opção de utilizar letras maiúsculas na composição do identificador, isso fornece uma maior possibilidade de identificadores, mas mantém o problema de previsibilidade, pois é só utilizar base 62 no lugar da base 36.

A solução para isso foi a geração de palavras utilizando a função rand()[17] da linguagem PHP. Considerando um grupo de caracteres da seguinte forma: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z].

Nossa rotina de geração de identificador não recebe nenhum argumento e para cada uma das cinco iterações gera um número inteiro aleatório entre 1 e 62 e concatena ao identificador o elemento do grupo de carácter pertencente à posição do número gerado. Após o fim das iterações teremos o identificador com 5 alfanuméricos, dificultando assim a previsibilidade das URLs.

Entrada: vazio

Saída : Combinação de 5 alfanuméricos aleatórios

$charset \leftarrow [0, 1, 2, ..., X, Y, Z]$;

para $i \leftarrow 1$ **até** 5 **faça**

  $identificador[i] \leftarrow charset[rand(1, 62)]$;

fim

retorna $identificador$

Algoritmo 1: Gera identificador aleatório

Ao utilizar um grupo de carácteres de 62 elementos e um identificador de 5 elementos temos a possibilidade de criar 62^5 identificadores aleatórios diferentes. Mesmo que o número de registros chegue perto dos 916.132.832, é possível aumentar o número do tamanho dos identificadores sem afetar as URLs já salvas. Apenas incrementando em um, já chegamos a 56.800.235.584 possibilidades.

3.7.2 Desativação de links

Foi necessário adicionar uma forma de desativar link, pois não era possível. A única alternativa era a exclusão no painel de administração, mas a exclusão permitia que o link fosse readicionado logo em seguida. Foi adicionado um campo booliano, chamado "Ativo" e é verificado seu conteúdo todas as vezes antes de redirecionar uma página. Esse

campo vem por padrão definido como verdadeiro e na identificação de uma URL indevida ele é mudado para falso.

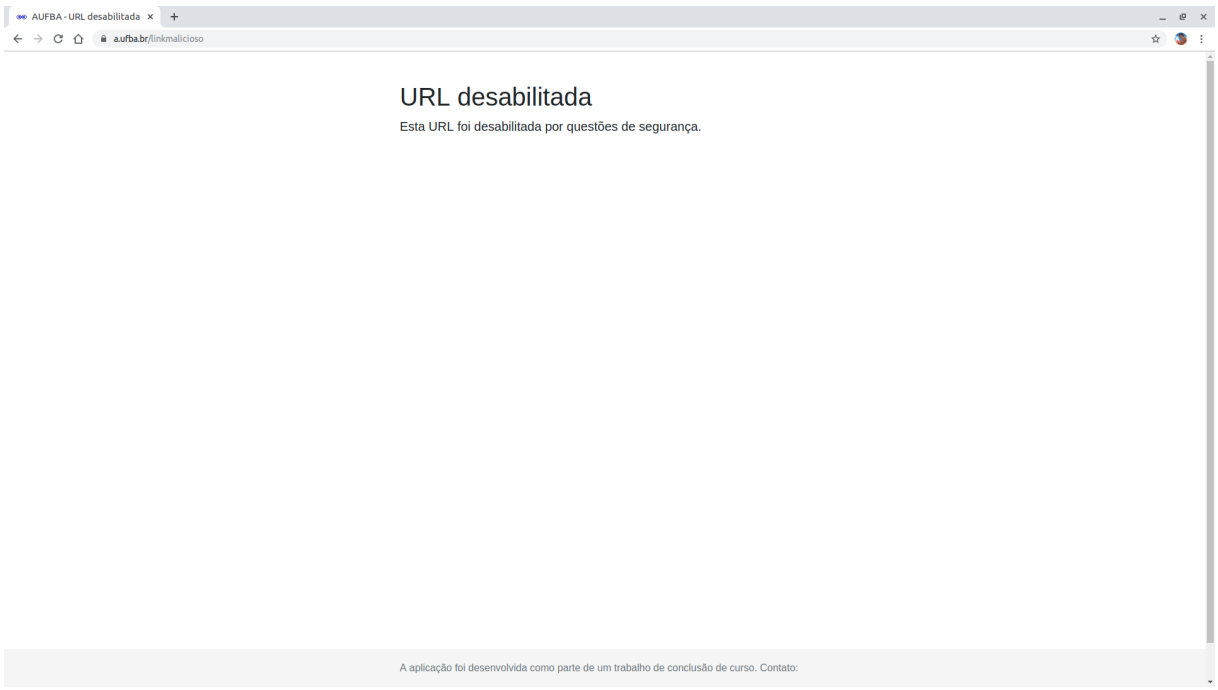
O processo de redirecionamento funciona da seguinte forma: Ao fornecer acessar uma URL no formato da seção 2.2 verifica se o identificador se encontra na base de dados, se a resposta for positiva é feito o redirecionamento para a URL longa vinculada a esse identificador.

Ao acessar uma URL encurtada a primeira vez era informado ao navegador que de agora em diante o recurso foi movida permanentemente para a URL longa. Isso é feito retornando o código de status 301 do protocolo http[7]. Dessa forma a partir da segunda requisição o navegador já para onde redirecionar sem consultar a aplicação e vai direto para o endereço em que foi redirecionado da primeira vez.

Esse comportamento é interessante pois evita uma requisição ao servidor, reduzindo o tempo de resposta do redirecionamento, porém nós queremos a opção de permitir a desativação de um link caso ele seja reportado como indevido. A aplicação passou a informar ao navegador que o recurso foi movido temporariamente e não mais permanentemente, através do código 302.

Feito isso ao requisitar uma URL à aplicação, após verificar se ela existe na base de dados verifica-se também se ela está ativa, se não estiver, é necessário informar ao usuário que ela está desativada. Foi adicionada uma página que traz essa informação.

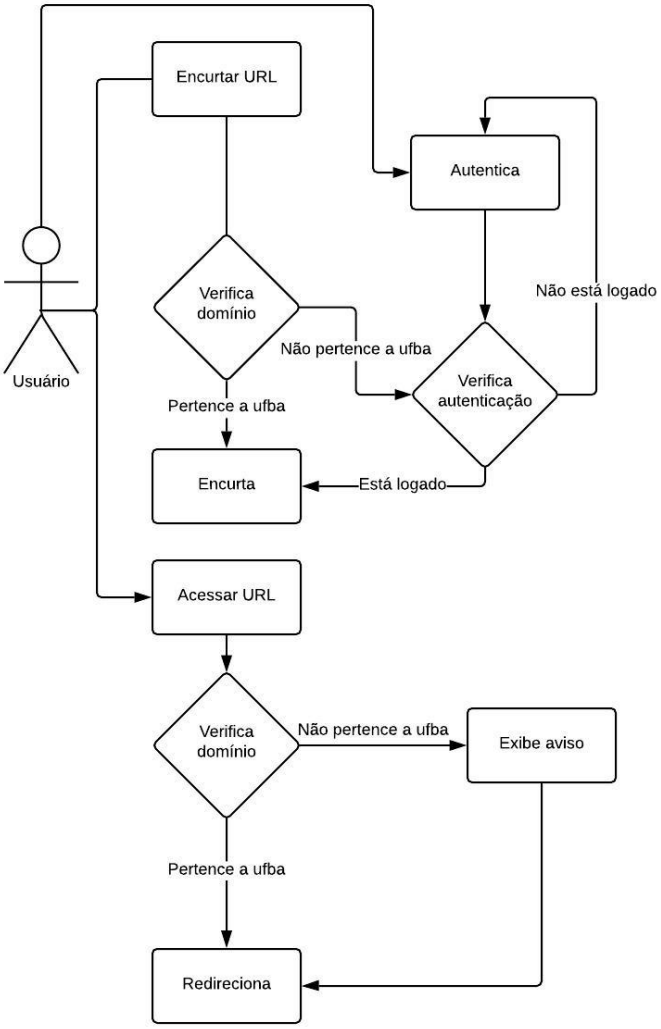
Figura 3.1: AUFBA - URL desabilitada



3.8 Visão Geral

As duas principais ações do usuário são encurtar uma URL ou acessar uma URL encurtada. Para a primeira ação basta o usuário acessar a página principal da aplicação[18] informar a URL longa e o identificador, que caso não informado é gerado automaticamente.

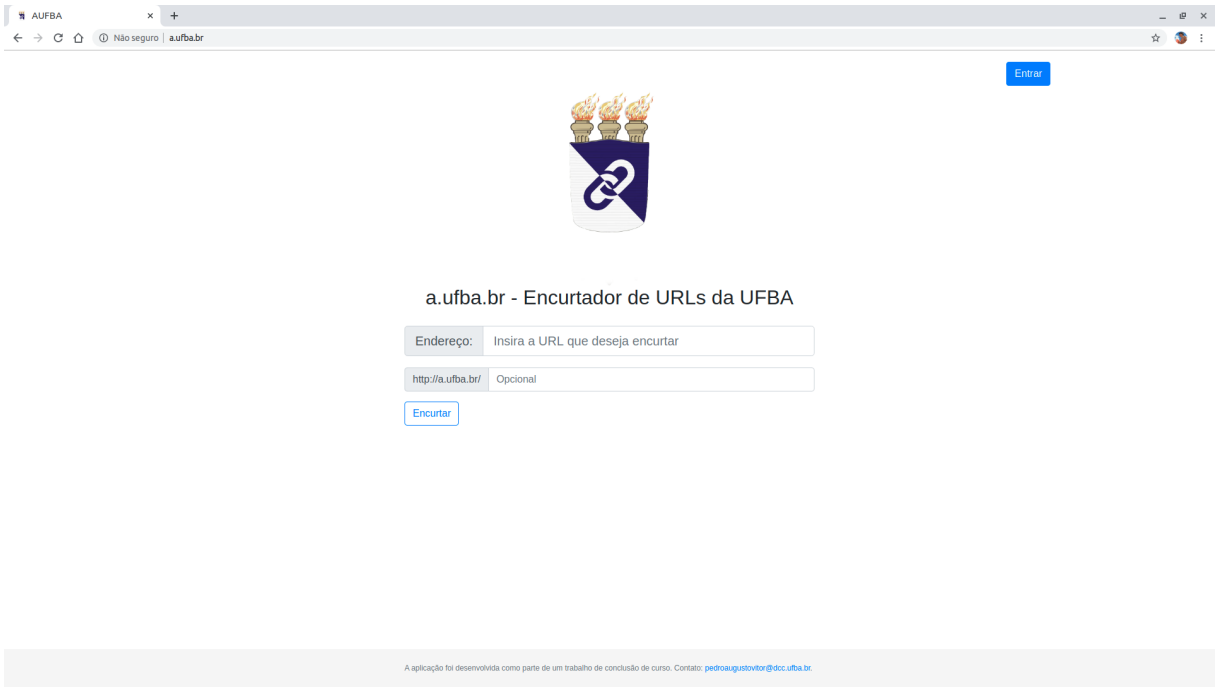
Figura 3.2: Fluxograma



3.8.1 Encurtar URL

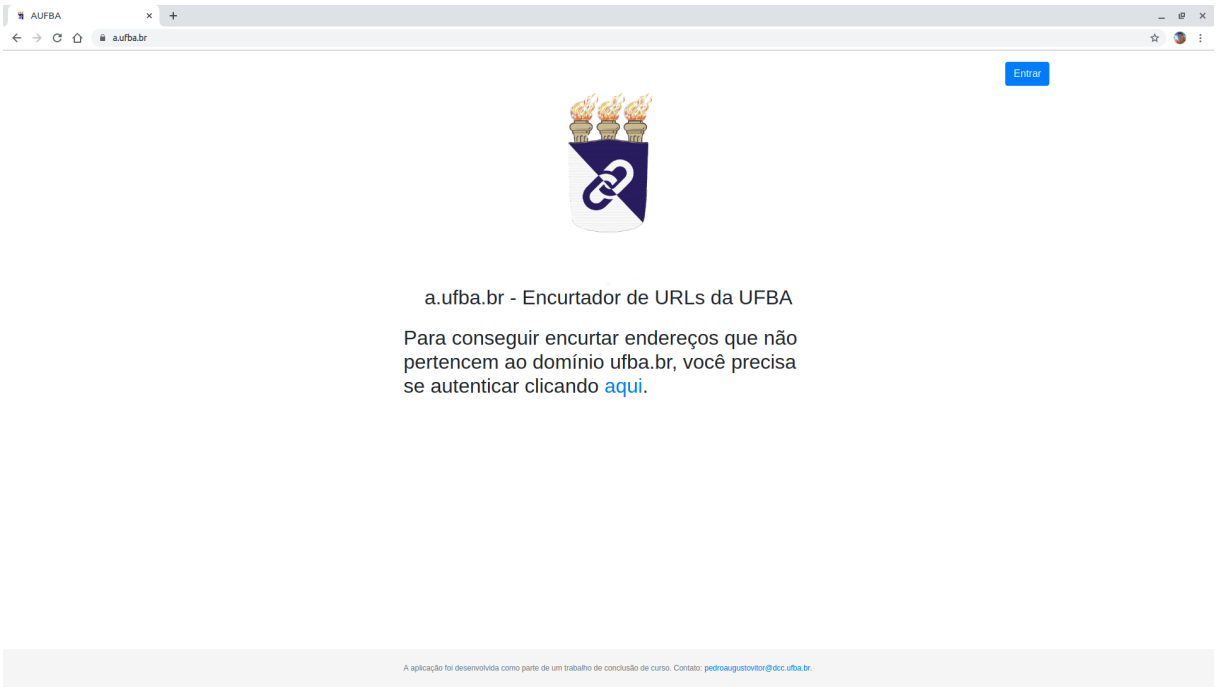
Esta é a tela principal da aplicação que permite encurtar uma URL:

Figura 3.3: AUFBA - Tela principal



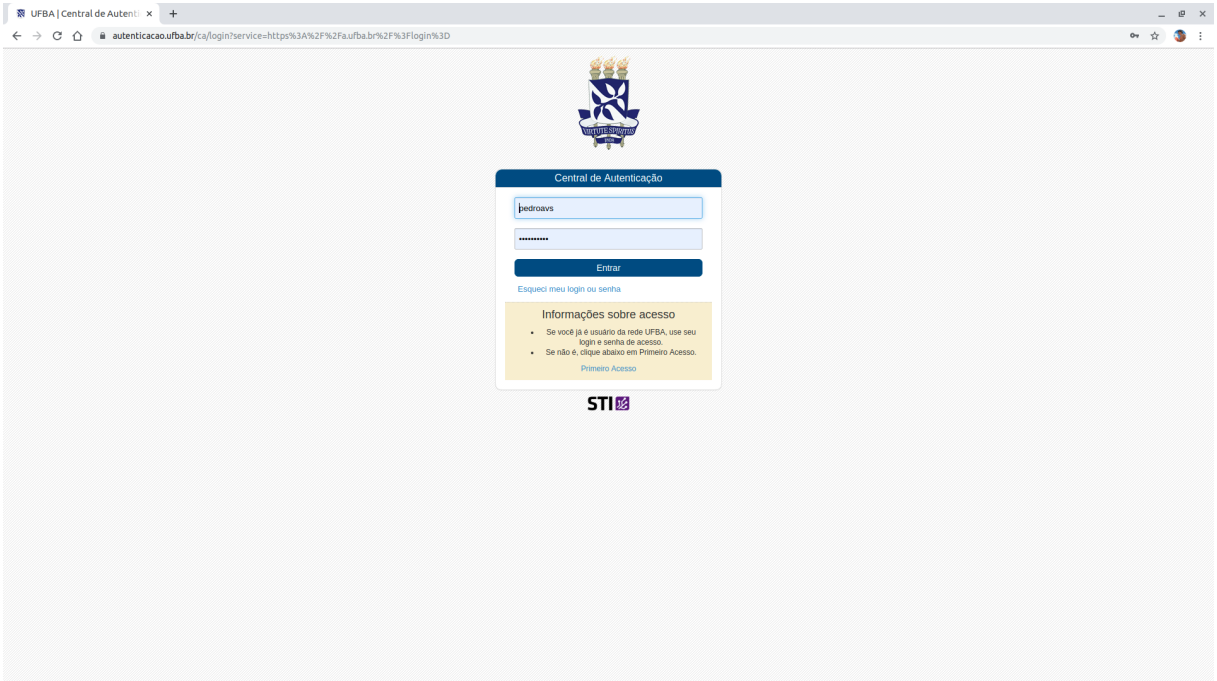
Observa-se que há opção de login na parte superior da tela, no botão: Entrar. Isso é por conta de o usuário ainda não estar autenticado. Após inserir as informações e clicar em encurtar, caso o usuário não esteja logado e a URL não pertença a o domínio UFBA, temos a seguinte tela:

Figura 3.4: AUFBA - URL não pertence ao domínio UFBA



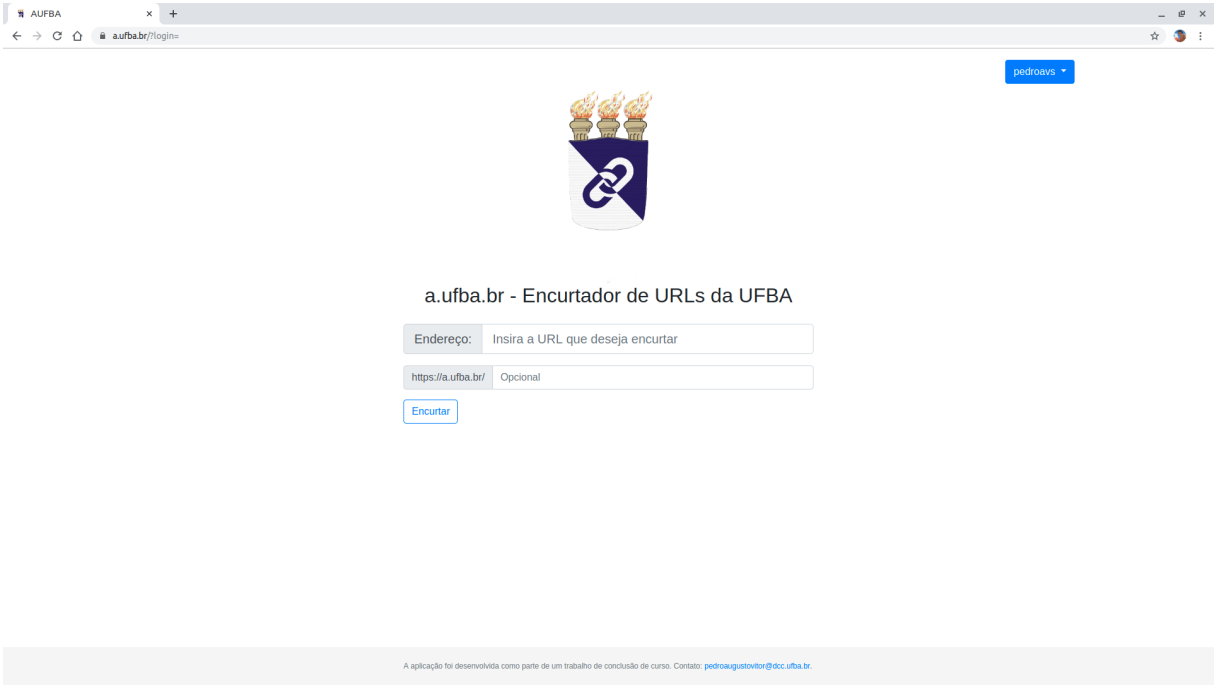
A mensagem "Para conseguir encurtar endereços que não pertencem ao domínio ufba.br você precisa se autenticar clicando aqui" é exibida e ao clicar no link informado o usuário é redirecionado para a página de autenticação da UFBA[11].

Figura 3.5: Autenticação UFBA - Tela de login



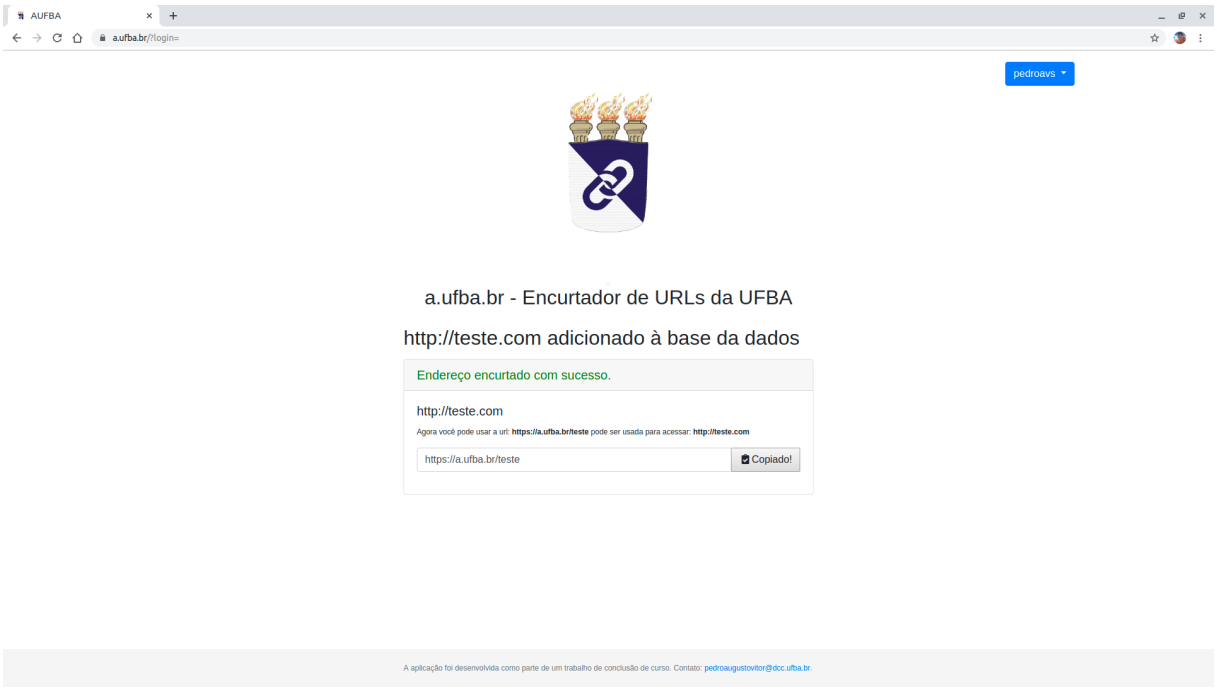
Após fornecer um usuário e senha válido e clicar em “Entrar”, retorna-se para a página principal AUFBA. Nesse momento o usuário está autenticado e é possível ver seu nome de usuário na parte superior da tela, onde antes ficava a opção “Entrar”.

Figura 3.6: AUFBA - Tela inicial pós autenticação



Como já definido, sendo uma URL pertencente ao domínio UFBA ou o se usuário estiver logado será exibida a seguinte tela após a tentativa de encurtamento:

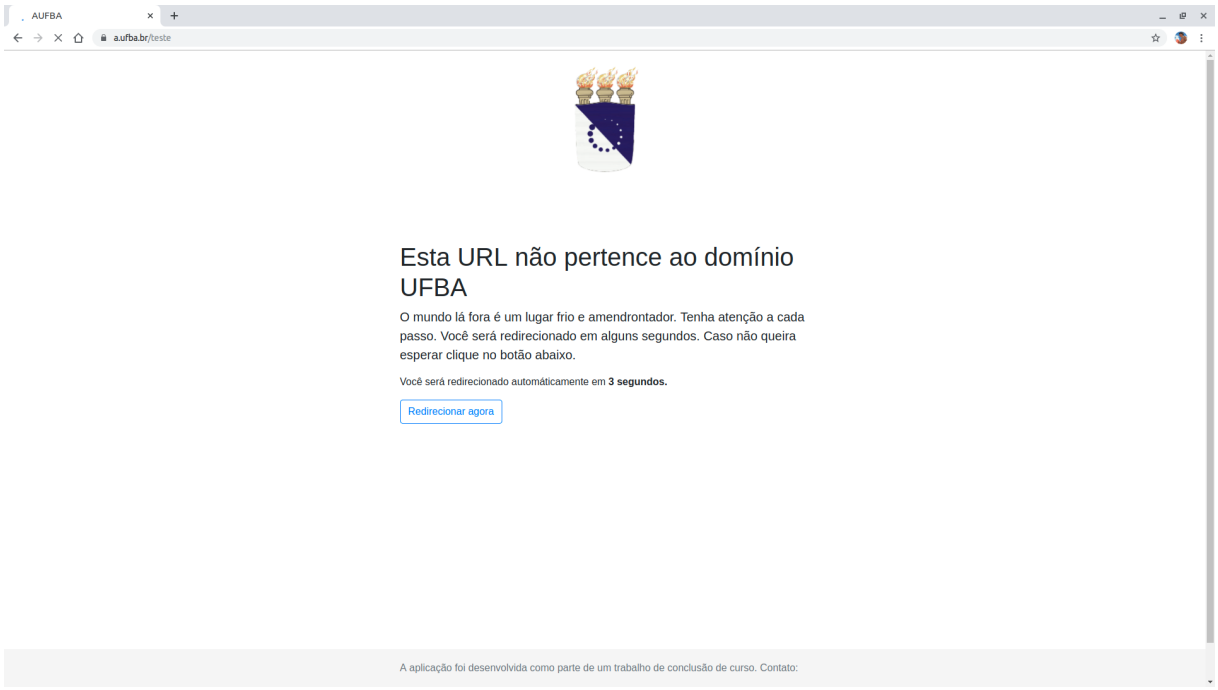
Figura 3.7: AUFBA - URL encurtada com sucesso



3.8.2 Acessando URL curta

Esse é um processo bem mais simples. Ao fazer uma requisição a uma URL curta é verificado o domínio vinculado da URL longe vinculada ao identificador. Caso o domínio pertença ao domínio da UFBA, a aplicação redireciona sem nenhuma tela intermediária. Por outro lado, se não pertencer é exibida a seguinte tela ao usuário:

Figura 3.8: AUFBA - Aviso



O usuário pode esperar 5 segundos ou clicar no botão “Redirecionar agora” e ser redirecionado imediatamente.

4. Conclusão

Este trabalho além de ter proposto uma solução para facilitar o acesso a determinados endereços, sem deixar de se preocupar com a segurança, pois ao vincularmos um serviço a instituição, requer que não haja o uso para fins que não condizem com as normas da mesma, permite também o avanço da solução para tratar maiores problemas.

Foi conseguido implementar a solução com todas as requisições inicialmente propostas, sendo elas autenticação utilizando o Autenticação UFBA, desativação de URLs encurtadas, controle das URLs que não pertencem ao domínio UFBA, desde a criação até o acesso. Tudo isso com uma interface simples, responsiva e que consegue suprir todas as necessidades destacadas dentro do escopo.

Observamos que grande parte das decisões tomadas durante o processo de implementação do serviço foi visando a adequação da plataforma dentro da infraestrutura fornecida pela UFBA, como por exemplo a escolha da linguagem de programação para a escrita da aplicação, a escolha de um projeto que permitisse o deploy com apenas os recursos disponibilizados pela STI.

Como houve a preocupação de não divulgar o serviço antes de grande parte das implementações serem finalizadas, por questões de segurança não foi possível analisar a utilização em larga escala do serviço, porém foi possível finalizar todo o trabalho sem essa análise por conta da simplicidade da solução, onde foi possível cobrir a maior parte do uso apena com testes.

Como trabalhos futuros tem-se o exemplo do Twitter que utilizou seu encurtador de URLs com o objetivo controlar da melhor forma a disseminação dos endereços dentro de sua plataforma, ao converter todos os links que circulam por ele em URLs curtas, assim que identificado a necessidade de se bloquear o acesso a determinada URL, com o simples ato de desativar o redirecionamento, permite-se isso. Desta forma abre-se a possibilidade de se utilizar a mesma solução dentro dos sistemas da UFBA, como por exemplo o servidor de e-mails convertendo URLs nos corpos das mensagens através do serviço desenvolvido nesse trabalho, depois de identificado uma URL maliciosa, desativando-a dentro do encurtador de URLs, impede-se o acesso por parte dos usuários. Outra possibilidade é a integração com catálogos de URLs, como por exemplo o CaUMa, verificando se as URLs longas estão nesses catálogos tanto no ato da inserção quando periodicamente, de forma que as ações sejam o mínimo possível percebidas pelo usuário do serviço e que a indisponibilidade desses serviços não impeçam o funcionamento do encurtador de URLs.

Referências

[1] Berners T. Lee, L. Masinter, and M. Mccahill. RFC 1738: Uniform resource locator (URL). http://www.ietf.org/rfc/rfc1738.txt, 1994.

[2] M Zainal Arifin. The influence of url shortener on pagerank. *IEESE International Journal of Science and Technology*, 2(2):16, 2013.

[3] Alexander Neumann, Johannes Barnickel, and Ulrike Meyer. Security and privacy implications of url shortening services. *Web 2.0 Security and Privacy 2011 Conference*, 2011.

[4] Twitter. O serviço de links do twitter. `https://help.twitter.com/pt/using-twitter/url-shortener`.

[5] Rogerio Bastos, Paula Tavares, Lucas Borges, Italo Brito, Edilson Lima, Liliana Solha, et al. Catálogo de fraudes e catálogo de urls maliciosas: Identificação e combate a fraudes eletrônicas na rede acadêmica brasileira. *Sexta Conferencia de Directores de Tecnología de Información, TICAL 2016*, 2016.

[6] Mary K. Taylor and Diane Hudson. "linkrot"and the usefulness of web site bibliographies. *Reference & User Services Quarterly*, 39(3):273–277, 2000.

[7] Mozilla Developer. Códigos de status de respostas http. `https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status`.

[8] Jonathan Zittrain, Kendra Albert, and Lawrence Lessig. Perma: Scoping and addressing the problem of link and reference rot in legal citations. *Legal Information Management*, 14(2):88–99, 2014.

[9] Sangho Lee and Jong Kim. Fluxing botnet command and control channels with url shortening services. *Computer Communications*, 36(3):320–332, 2013.

[10] Wikipedia. Spam blacklist. `https://meta.wikimedia.org/wiki/Spam_blacklist`.

[11] UFBA. Autenticação. `https://autenticacao.ufba.br/`.

[12] Federico Maggi, Alessandro Frossi, Stefano Zanero, Gianluca Stringhini, Brett Stone-Gross, Christopher Kruegel, and Giovanni Vigna. Two years of short urls internet measurement: Security threats and countermeasures. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 861–872, New York, NY, USA, 2013. ACM.

[13] Yourls. Site oficial. `https://https://yourls.org/`.

[14] Polr. Site oficial. `https://polrproject.org/`.

[15] Polr. Demo. `https://demo.polr.me/`.

[16] Wikipedia. Sistemas de numeração hexatrigesimal. `https://pt.wikipedia.org/wiki/Sistema_de_numera%C3%A7%C3%A3o_hexatrigesimal`.

[17] PHP. Função randômica. `https://www.php.net/manual/pt_BR/function.rand.php`.

[18] AUFBA. O encurtador de urls da ufba. `https://a.ufba.br`.

[19] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1245–1254, New York, NY, USA, 2009. ACM.

[20] Demetris Antoniades, Iasonas Polakis, Georgios Kontaxis, Elias Athanasopoulos, Sotiris Ioannidis, Evangelos P. Markatos, and Thomas Karagiannis. We.b: The web of short urls. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 715–724, New York, NY, USA, 2011. ACM.