



# Avaliação de Notícias

Supervisor: Paulo Rurato

Orientador: Luís Borges Gouveia

Alunos:

Miguel Ferreira nº 27256

Tiago Cardoso nº 28106

Universidade Fernando Pessoa

Maio de 2017

# Índice

Introdução .....	2
Objetivos .....	3
Qualidade dos Resultados .....	4
Arquitetura da Aplicação .....	5
Requisitos Funcionais .....	6
Requisitos Não-Funcionais .....	7
Mapa do Site .....	8
Planeamento do Projeto .....	9

# Introdução

O presente trabalho é sobre avaliação de notícias falsas, mais concretamente uma aplicação para detectar e alertar o utilizador acerca da veracidade das mesmas.

O objetivo deste trabalho é evitar que cada um de nós, sirva de caixa de ressonância ao partilhar e assim tornar mais credível, notícias que são falsas, fabricadas, desatualizadas, parciais e que em vez de servir para informar, servem interesses não especificados na própria notícia.

As principais tecnologias utilizadas para desenvolver esta aplicação foram PHP (Framework Laravel), Javascript, HTML e CSS.

# Objetivos

Dado o estado da comunicação social nos tempos que decorrem e da quantidade de notícias falsas que circulam na internet e principalmente nas redes sociais onde habitualmente não faltam debates sobre este assunto nos comentários, o objetivo deste trabalho é de criar uma aplicação que seja uma possível referencia para quem procura saber se uma determinada notícia é verdadeira ou falsa. Resumidamente, a nossa aplicação compromete-se em armazenar numa base de dados as notícias que vão sendo pesquisadas e consequentemente votadas pelos utilizadores que visitem e utilizem a nossa aplicação. Há medida que o número de votações aumenta, vai permitir que a longo prazo a partir de uma opinião generalizada possamos saber se a notícia é ou não credível.

Depois de alguma pesquisa e de nos reunirmos com o coordenador do projeto, chegamos ao consenso de que esta seria a solução mais eficaz para resolver este problema, devido ao facto de não termos conhecimento sobre machine learning ou inteligência artificial que seria a solução ideal para resolver este problema.

# Qualidade dos Resultados

A nossa aplicação representa uma possível solução entre várias a um problema que infelizmente é recorrente, e que com as redes sociais tem tendência a crescer. Contudo, não dominando sobre o assunto de inteligência artificial (bots) para podermos comparar resultados entre várias páginas de uma forma mais eficaz, tivemos que nos restringir a um conjunto de soluções.

Solucionamos então o nosso problema da seguinte forma. O utilizador ao introduzir um link de uma noticia no web site ou na extensão Chrome, o título desse link e respectivo conteúdo da noticia vai ser analisado. Como o API de noticias da Google não está disponível para acesso livre, tivemos que utilizar a API de noticias do motor de pesquisa BING que nos deu garantias, e apesar de só estar disponível para uso gratuito durante 90 dias, não comprometeu o trabalho final. Depois de retornados os resultados, a aplicação vai mostrar a lista de web sites onde foram encontradas noticias com títulos semelhantes, e em alguns casos até iguais ao link da noticia que foi introduzido inicialmente.

Em seguida, ponderamos em atribuir uma percentagem à quantidade de noticias retornadas com um título semelhante, mas desta forma não sendo 100% eficaz poderíamos estar a induzir o utilizador em erro. Junto com o professor orientador ficou decidido que a melhor solução seria atribuir um peso à noticia introduzida através de uma votação e assim fizemos. Para terminar, o utilizador pode votar numa noticia e esse voto fica registado na base de dados, aumentando assim a credibilidade da mesma. Os administradores da aplicação tem como responsabilidade verificar as noticias que vão sendo inseridas na base de dados e marcar as mesmas como “verificadas”. Ao efetuar esta ação o utilizador será avisado antes de votar na noticia.

# Arquitetura da Aplicação

A arquitetura ou modelo utilizado para desenvolver a aplicação foi o MVC que em inglês significa, Model View e Controller. É um padrão de arquitetura de software (design pattern) que separa a representação da informação da interação do utilizador com ele. O modelo consiste nos dados da aplicação, regras de negócios, lógica e funções. Uma view pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões do mesmo dado, como um gráfico de barras e uma visão tabular para contadores. O controlador faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão. As ideias centrais por trás do MVC são a reusabilidade de código e separação de conceitos.

Além de dividir a aplicação em três tipos de componentes, o MVC define as interações entre eles. O controlador envia comandos para o modelo para atualizar o seu estado. O controlador também pode enviar comandos para a view associada para alterar a apresentação da visão do modelo. Um modelo armazena dados e notifica suas views e controladores associados quando há uma mudança do seu estado. Estas notificações permitem que as views produzam saídas atualizadas e que os controladores alterem o conjunto de comandos disponíveis. A view gera uma representação dos dados presentes no modelo solicitado.

# Requisitos Funcionais

Existem dois tipos de contas associadas à nossa aplicação: administrador e visitante.

Requisitos funcionais do **visitante**:

- Inserir um link válido de uma notícia.
- Votar numa notícia consoante os resultados obtidos pela aplicação.

Nota: Estas duas ações poderão ser feitas através do Website ou da extensão para o Chrome.

Requisitos funcionais do **administrador**:

- Adicionar ou remover um ou vários administradores.
- Verificar, pesquisar ou remover noticiais que vão sendo inseridas na base de dados, à medida que os visitantes votam nas mesmas.

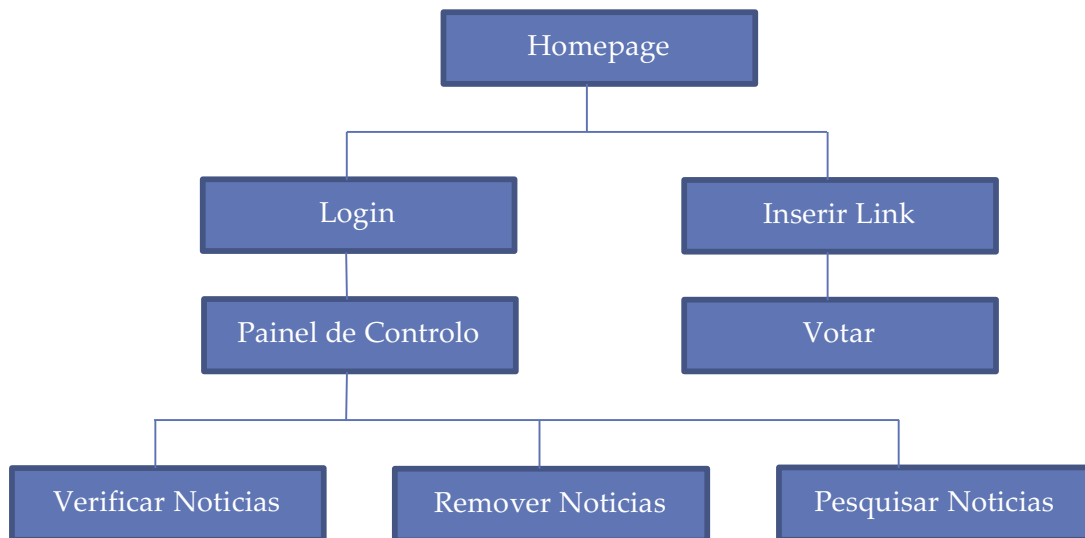
# Requisitos Não-Funcionais

- Para aceder à aplicação, é necessário que o utilizador esteja conectado à rede.
- O padrão de arquitetura de software utilizado será MVC com recurso à framework Laravel.
- A aplicação deverá ser fácil de usar e que dispense muitos recursos gráficos.
- A base de dados deve ser protegida para acesso apenas a utilizadores autorizados, neste caso aos administradores.
- A aplicação deverá garantir que não haja perda de dados, e deve ser capaz de tratar exceções e de recuperar de falhas.
- Quando um visitante insere um link de uma notícia, o tempo de resposta da aplicação não deve ultrapassar os 20 segundos, caso contrário a pesquisa não será bem sucedida.
- A interface da aplicação deverá comporta-se adequadamente independente do front-end que será utilizado para acesso – Browser, Smartphone ou Tablet.

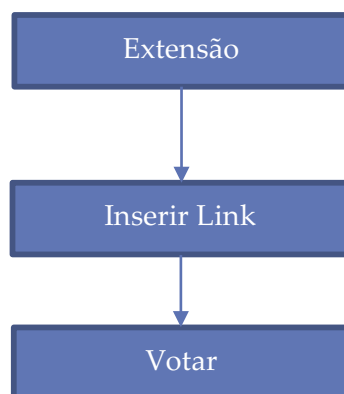


# Mapa do Site

## Website:



## Extensão Chrome:



# Planeamento do Projeto

**Segue em anexo**, um documento que contém o planeamento do projeto de acordo com o que foi feito ao longo do semestre.