

# **Instituto Federal de São Paulo - *campus* Barretos**

## **Pós-Graduação *lato senso* em Programação para *Internet* e Dispositivos Móveis**

### **Desenvolvimento para *Internet* II**

#### *Avaliação*

#### **Introdução**

Nesta avaliação, você deverá desenvolver um sistema para um *blog* de notícias, de acordo com as especificações e restrições listadas neste documento.

#### **Forma de Avaliação**

A avaliação levará em consideração as seguintes características do sistema desenvolvido:

- Correto atendimento do sistema em relação às especificações fornecidas neste documento;
- Utilização correta dos conceitos trabalhados na disciplina;
- Funcionamento correto, eficiente e seguro do sistema desenvolvido;
- Correta explanação dos conceitos utilizados no sistema;
- Correta explanação do funcionamento do sistema.

#### **Formação de Grupos**

Esta avaliação pode ser feita de forma individual ou em dupla, a critério dos discentes.

#### **Data de Entrega**

O prazo máximo para a entrega do código-fonte do sistema desenvolvido é o dia 19/abr/2023 às 23:59.

#### **Forma de Entrega**

A entrega deverá ser realizada em atividade cadastrada no Moodle institucional, disponível em <https://moodle.brt.ifsp.edu.br/>. O código-fonte deve estar em repositório público no GitLab ou GitHub, sendo o seu endereço informado no Moodle institucional.

# 1 Descrição geral do sistema

O sistema a ser desenvolvido nesta avaliação deve ser um sistema de um *blog* de notícias, em que diferentes autores podem publicar suas notícias. Essas podem ser visualizadas por leitores, que podem interagir com elas através de reações e de comentários.

## 1.1 Motivação

A *internet* é uma fonte atual muito importante para a divulgação de notícias, por fornecer uma forma rápida e prática para a disseminação de informações. Uma pessoa conectada à *internet* pode visualizar notícias de diversas fontes, bem como autores podem facilmente escrever suas notícias de qualquer aparelho que possua uma conexão com a *internet* e um navegador atualizado.

## 1.2 Objetivos

O objetivo principal deste sistema é ser um sistema gratuito de fácil utilização, tanto por parte dos autores, na escrita de suas notícias, bem como ser uma fonte de leitura prática para visitantes do sistema, permitindo que eles possam interagir com as notícias publicadas.

## 2 Requisitos Funcionais

Os requisitos funcionais descrevem o comportamento do sistema e estão listados aqui em ordem aleatória. Todo o comportamento descrito aqui é obrigatório e deve ser realizado. Incrementos podem ser feitos, a critério do(s) discente(s), desde que não alterem o comportamento dos requisitos.

### 2.1 RF01 - Tela inicial

O sistema deve exibir, na sua tela inicial, uma lista de notícias, exibindo seu título e seu subtítulo, bem como a data da sua publicação. Elas devem ser ordenadas da notícia mais atual para a mais antiga. Ao clicar no título de uma notícia, a mesma deve ser exibida (ver RF02).

### 2.2 RF02 - Visualização de notícia

Ao exibir uma notícia, deve-se exibir seu título, subtítulo, data da publicação, nome do autor da notícia e o texto da notícia. Exibir também a quantidade de leitores que gostaram da notícia, a quantidade de leitores que não gostaram da notícia, e os comentários, se houverem.

### 2.3 RF03 - Cadastro de leitores

Qualquer pessoa pode ser cadastrar como leitor de notícias do sistema, informando seu nome, e-mail (único), data de nascimento e uma senha forte. Após informar esses dados, o sistema deve enviar um e-mail para o e-mail cadastrado, informando ao usuário um código de ativação, que deve ser informado no sistema para completar seu cadastro. Leitores que não tiverem confirmado o código de ativação não podem usar as funcionalidades do sistema.

### 2.4 RF04 - Manutenção de autores

O cadastro de autores é feito através do usuário administrador, que é cadastrado na primeira utilização do sistema (ver RF07). De cada autor armazena-se seu nome, e-mail, um apelido e a data de nascimento. No cadastro de um autor, a senha de entrada no sistema é enviada a ele em seu e-mail, devendo ser alterada após a primeira autenticação no sistema (ver RF06). O usuário administrador pode editar apenas o nome de um autor, ou então bloquear seu acesso ao sistema. Autores só podem ser excluídos do sistema caso eles não tenham nenhuma notícia publicada.

### 2.5 RF05 - Manutenção de usuários

Todos os usuários (leitores e autores) podem alterar seu e-mail (desde que o novo e-mail seja único no sistema) e data de nascimento. Caso o usuário altere seu e-mail, é necessário confirmar o novo e-mail com um código enviado ao novo e-mail.

### 2.6 RF06 - Redefinição de senha

Para que um usuário (leitor ou autor) possa alterar sua senha, é necessário que o mesmo faça essa solicitação através do sistema. Um e-mail deve ser enviado ao e-mail do usuário com um código, que deve ser utilizado para a criação da nova senha. A nova senha não deve ser igual à última senha. Utilizar boas práticas para criação e armazenamento de senhas.

## 2.7 RF07 - Primeira execução

Na primeira execução do sistema, deve ser solicitado o nome, e-mail e a senha do usuário administrador. Este é o único usuário administrador que o sistema deve possuir. Para alterar os dados do administrador, veja o RF05. Para alterar a senha do administrador, veja o RF06.

## 2.8 RF08 - Reação de leitores a uma notícia

Um leitor pode reagir a uma notícia informando que gostou ou não gostou dela. Caso essa informação tenha sido armazenada, destacar quando esse leitor visualizar essa notícia. Apenas leitores com cadastro confirmado (e não bloqueado) no sistema podem realizar esta operação. É possível para o leitor remover essa informação, caso ela tenha sido informada.

## 2.9 RF09 - Comentários de uma notícia

Um leitor pode comentar uma notícia, informando um texto a ser exibido no final da página da mesma, em ordem cronológica. O comentário não pode ser alterado pelo leitor. Apenas leitores com cadastro confirmado (e não bloqueado) no sistema podem realizar esta operação. Não é possível para o leitor remover um comentário realizado.

## 2.10 RF10 - Manutenção de notícia

Um autor pode criar uma nova notícia, informando seu título, subtítulo e o texto da notícia. Uma notícia não é exibida na lista de notícias após ser salva, ela deve ser publicada para tal. Uma notícia não publicada pode ter todas as suas informações editadas. A um autor é permitido alterar apenas o texto de uma notícia que ele tenha publicado. Cada autor pode alterar apenas textos de suas próprias notícias. A última data de alteração da notícia já publicada deve ser informada junto com a data de publicação da notícia, quando existir. Uma notícia pode ser removida apenas se ela não foi publicada.

## 2.11 RF11 - Exibição de notícias de autor

Em alguma seção do sistema, deve-se exibir (para os leitores) todos os autores não bloqueados presentes no sistema. Deve-se exibir o nome deles, a quantidade de notícias que cada um escreveu e um *link* para suas notícias, em que apenas as notícias do autor selecionado devem ser exibidas. Este *link* deve utilizar o apelido cadastrado do autor como rota. Por exemplo: `/noticias/tiago` exibirá apenas as notícias do autor com apelido **tiago**.

## 2.12 RF12 - Manutenção de leitores

O administrador do sistema pode bloquear o acesso de leitores, de acordo com seus critérios. Leitores bloqueados não podem acessar o sistema, mas possuem comentários e reações a notícias exibidas no sistema.

## 2.13 RF13 - Manutenção de comentários

O administrador do sistema pode excluir comentários a seu critério.

## 3 Requisitos Não Funcionais

Os requisitos não funcionais descrevem qualidades e características do sistema. Todas os requisitos não funcionais descritos aqui são obrigatórios e devem ser cumpridos no desenvolvimento do sistema.

### 3.1 RNF01 - Plataforma e uso esperado do sistema

O sistema deve ser acessível por navegadores atuais, sendo baseado em HTML5 e Javascript. Todas as tecnologias utilizadas no *frontend*, *backend*, servidor, servidor de e-mail, entre outros, devem ser gratuitas. O *site* deverá ser possível de ser hospedado em servidor com o sistema operacional Linux.

### 3.2 RNF02 - Arquitetura do sistema

O sistema deve utilizar uma arquitetura *Model-View-Controller* (MVC), sendo a camada responsável pelas *views* desacoplada das demais camadas (ver RNF03 e RNF04). As camadas referentes às *models* e *controllers* devem ser acessadas através de uma *Application Programming Interface* (API) (ver RNF05).

### 3.3 RNF03 - Características do *frontend*

A interface do sistema deve ser responsiva, utilizando alguma biblioteca gratuita para criação da *interface* com o usuário (Bootstrap, Materialize, Tailwind, Flowbite, etc.). O sistema deve ser construído utilizando o conceito de *Single Page Application* (SPA). O único carregamento de página aparente para o usuário deve acontecer ao acessar o sistema. Todas as trocas de "página" devem ocorrer utilizando rotas com nomes amigáveis e condizentes com a "página" acessada.

### 3.4 RNF04 - Características do *backend*

A camada *View* do sistema deve ser totalmente escrita utilizando alguma biblioteca/*framework* Javascript descritiva(o), como React.js, Vue.js, Svelte, etc. Não mesclar a tecnologia escolhida com outra tecnologia para criação de *Views* (PHP, Java ou C, por exemplo). O sistema deve ser dinâmico, com troca de componentes realizada pela tecnologia escolhida, e não por Javascript. Para obter informações sobre dados, o sistema deve acessar uma API (ver RNF05).

### 3.5 RNF05 - Características da API

A API para controle de informações deve ser uma API RESTful, escrita em linguagem de livre escolha (C#, Java, PHP, Python, Javascript, etc.). Ela deve poder ser acessada apenas pelo sistema criado e seguir o padrão *Controller/Model* do padrão MVC. Os dados de armazenamento deve ser acessados apenas pela API utilizando um requisições assíncronas e uma biblioteca *Object Relational Mapping* (ORM). A API deve ser segura, permitindo acesso a seus métodos através de autenticação e autorização (com exceção do método de autenticação/autorização).

### 3.6 RNF06 - Características de armazenamento

O armazenamento do sistema deve possuir algumas características básicas, como acesso por múltiplos usuários, controle de concorrência, atomicidade das operações, recuperação dos dados e durabilidade dos dados (ou seja, mesmo se uma falha ocorrer no sistema, os dados devem

ser preservados e estar em perfeito estado para serem utilizados). Os dados devem ser acessados apenas após autenticação e autorização de usuário, sendo que informações sensíveis, se houverem, devem ser armazenadas de forma criptografada. Deve ser possível criar ou restaurar *backup* de maneira rápida e eficiente (utilização de, no máximo, dois comandos para executar essas tarefas).