

## **1. Fase de Concepção / Iniciação**

### **1.1. Quais são as principais funcionalidades que a API deve oferecer?**

#### **Funcionalidades Detalhadas do Blog**

##### **1. Sistema de Login e Cadastro de Usuários**

**Descrição:** Permitir que os usuários se cadastrem e façam login para acessar funcionalidades adicionais.

##### **Funcionalidades:**

- **Cadastro de Usuário:** Formulário para novos usuários se registrarem com campos como nome, e-mail, senha, e confirmação de senha.
- **Login:** Tela de login onde usuários inserem e-mail e senha para acessar a área restrita do blog.
- **Recuperação de Senha:** Opção para o usuário recuperar a senha caso a esqueça, geralmente via e-mail.
- **Perfil de Usuário:** Página onde o usuário pode visualizar e editar suas informações pessoais.
- **Tecnologias Envolvidas:** PHP (para lógica de backend), JAVA(Springboot) para API, MySQL (para armazenamento de dados), HTML/CSS/JavaScript (para formulários e interface) e Bootstrap (para responsividade e reaproveitamento de componentes).

##### **2. Interface de Administração para Publicação de Artigos**

**Descrição:** Criar uma área restrita para administradores ou editores onde eles possam gerenciar o conteúdo do blog.

##### **Funcionalidades:**

- **Dashboard de Administração:** Tela inicial da área administrativa com resumo das atividades (número de artigos, comentários recentes, etc.).
- **Criação e Edição de Artigos:** Formulário para criação de novos artigos com campos como título, conteúdo, categoria, e upload de imagens. Inclui opção para editar ou deletar artigos existentes.
- **Gerenciamento de Categorias:** Permitir ao administrador criar, editar e excluir categorias que organizam os artigos.
- **Moderação de Comentários:** Visualizar e moderar comentários feitos pelos usuários nos artigos, com a opção de aprovar ou deletar.
- **Tecnologias Envolvidas:** PHP (para lógica de backend), JAVA(Springboot) para API, MySQL (para armazenar artigos e categorias), TinyMCE ou outro editor WYSIWYG (para formatação de texto), HTML/CSS/JavaScript (para a interface), Bootstrap (para responsividade e reaproveitamento de componentes).

##### **3. Publicação e Formatação de Artigos**

**Descrição:** Permitirá que os artigos sejam publicados com formatação rica, incluindo texto, imagens e links.

**Funcionalidades:**

- **Editor de Texto Rich Text:** Um editor WYSIWYG que permita aos administradores formatar o texto (negrito, itálico, listas, links, etc.).
- **Upload de Imagens:** Funcionalidade para inserir imagens diretamente no corpo do artigo, com possibilidade de redimensionamento.
- **Categorias e Tags:** Associa os artigos a uma ou mais categorias e permite a adição de tags para facilitar a busca.
- **Tecnologias Envolvidas:** PHP (para processamento de formulários e upload de arquivos), JAVA(Springboot) para API, MySQL (para armazenar dados dos artigos), HTML/CSS (para a exibição dos artigos), JavaScript (para interatividade no editor).

#### **4. Busca e Filtragem de Artigos**

**Descrição:** Oferecer aos visitantes e usuários cadastrados a capacidade de buscar artigos e filtrá-los por categorias ou tags.

**Funcionalidades:**

- **Barra de Busca:** Campo de busca na interface do blog que permita aos usuários procurar artigos por palavras-chave.
- **Filtragem por Categorias:** Funcionalidade para filtrar os artigos exibidos por uma ou mais categorias selecionadas.
- **Filtragem por Tags:** Similar à filtragem por categorias, mas utilizando tags atribuídas aos artigos.
- **Ordenação de Resultados:** Permitir que os usuários ordenem os resultados por data, relevância, ou popularidade.
- **Tecnologias Envolvidas:** PHP (para realizar consultas ao banco de dados), JAVA(Springboot) para API, MySQL (para armazenar e buscar os artigos), JavaScript (para atualizações dinâmicas de resultados), Bootstrap (para responsividade e reaproveitamento de componentes).

#### **5. Sistema de Comentários**

**Descrição:** Permitir que os usuários comentem nos artigos, promovendo a interação entre os leitores.

**Funcionalidades:**

- **Formulário de Comentários:** Campo para que usuários autenticados deixem comentários em um artigo.
- **Moderação de Comentários:** Área de administração onde os comentários podem ser aprovados, editados ou excluídos.

- **Respostas a Comentários:** Permitir que os usuários respondam aos comentários de outros, criando uma árvore de conversação.
- **Notificações:** Opcionalmente, notificar o autor do artigo ou o usuário quando seu comentário receber uma resposta.
- **Tecnologias Envolvidas:** PHP (para gerenciamento de comentários), JAVA(Springboot) para API, MySQL (para armazenar e consultar os comentários), HTML/CSS/JavaScript (para o frontend), Bootstrap (para responsividade e reaproveitamento de componentes).

## 6. Favoritos e Interação do Usuário

**Descrição:** Funcionalidade que permite aos usuários cadastrados salvar artigos como favoritos para leitura futura.

**Funcionalidades:**

- **Adicionar aos Favoritos:** Botão em cada artigo para permitir que o usuário o salve como favorito.
- **Gerenciamento de Favoritos:** Página no perfil do usuário onde ele pode ver e gerenciar seus artigos favoritos.
- **Tecnologias Envolvidas:** PHP (para salvar e recuperar favoritos), JAVA(Springboot) para API, MySQL (para armazenar informações de favoritos), HTML/CSS/JavaScript (para a interface do usuário), Bootstrap (para responsividade e reaproveitamento de componentes).

### 1.2. Quem são os stakeholders envolvidos, e quais são suas expectativas?

- **Pergunta:** Quem são as partes interessadas no projeto, e o que elas esperam da API?
  - Usuários finais que desejam acesso rápido e seguro ao conteúdo; a equipe de frontend que precisa de uma API estável e bem documentada.

### 1.3. Quais são os riscos potenciais que você antecipa?

- **Pergunta:** Que riscos você vê no desenvolvimento da API?
  - Exemplo: Falhas de segurança, problemas de performance, dificuldades na integração com o frontend, dificuldades no deploy.

### 1.4. Qual é o escopo inicial do projeto?

- **Pergunta:** Quais funcionalidades ou módulos são prioritários e devem ser abordados na primeira versão da API?
  - Foco inicial em CRUD de artigos e autenticação de usuários, deixando comentários e gestão de categorias para uma fase posterior.

### 1.5. Quais são as estimativas de custo e prazo para o desenvolvimento?

- **Pergunta:** Quanto tempo e quais recursos você acha que serão necessários para desenvolver a API?

- Ainda não sei sobre isso, preciso de um backend com banco de dados para a API, acho que Heroku, de preferência gratuito.
- Sobre o tempo 6 semanas para a API completa.

## Resposta 1(descartada)

Com as respostas da **Fase de Concepção / Iniciação** claras, vamos organizar o próximo passo e seguir para a **Fase de Elaboração**, onde iremos detalhar a modelagem da API e a mitigação dos riscos identificados.

### 2. Fase de Elaboração

Nesta fase, você começará a modelar a API de forma mais detalhada, além de revisar e mitigar os riscos. Vamos trabalhar com perguntas que orientam essa fase.

#### 2.1. Como será o modelo de dados da API?

- **Pergunta:** Como os dados serão organizados? Quais tabelas e relações entre elas serão necessárias? Pense nos principais componentes (usuários, artigos, categorias, comentários, favoritos).
  - Exemplo:
    - Tabela de usuários (id, nome, email, senha, perfil)
    - Tabela de artigos (id, título, conteúdo, categoria\_id, data\_publicação)
    - Tabela de categorias (id, nome)
    - Tabela de comentários (id, artigo\_id, usuário\_id, texto, data\_comentário)
    - Tabela de favoritos (id, artigo\_id, usuário\_id)

#### 2.2. Qual arquitetura será adotada para a API?

- **Pergunta:** Você mencionou usar Java com Spring Boot, mas como será o design da API? Pretende adotar o padrão RESTful? Quais endpoints serão definidos?
  - Exemplo: Arquitetura RESTful, endpoints organizados como:
    - /api/v1/artigos para CRUD de artigos
    - /api/v1/usuarios para gerenciamento de usuários
    - /api/v1/categorias para gestão de categorias
    - /api/v1/comentarios para sistema de comentários

### 2.3. Como você pretende mitigar os riscos identificados?

- **Pergunta:** Como planeja mitigar os riscos de segurança, performance e integração?
  - Exemplo:
    - **Segurança:** Implementação de autenticação JWT, criptografia de senhas.
    - **Performance:** Uso de caching em consultas frequentes e limitação de requisições por minuto (rate limiting).
    - **Integração:** Definição de padrões de comunicação clara com a equipe de frontend, documentação detalhada da API.

### 2.4. Quais tecnologias e frameworks serão utilizados?

- **Pergunta:** Além de Spring Boot, que outras ferramentas pretende usar?
  - Exemplo:
    - Spring Security para autenticação
    - JPA/Hibernate para mapeamento objeto-relacional (ORM)
    - Heroku para hospedagem
    - MySQL como banco de dados

### 2.5. Como será feito o protótipo da API?

- **Pergunta:** Quais endpoints serão prototipados inicialmente para testar a estrutura?
  - Exemplo: Prototipar inicialmente o CRUD de artigos e a autenticação de usuários.

---

Essas são as principais questões para a **Fase de Elaboração**. Ao responder essas perguntas, você estará mais próximo de uma estrutura sólida para o desenvolvimento. Vamos seguir essa fase?