Plano de Desenvolvimento da Aplicação de Bloco de Notas

1. Escopo do Projeto

1.1. Objetivo do Projeto Desenvolver uma aplicação de bloco de notas que permita aos usuários criar, editar e armazenar notas de forma segura. A aplicação incluirá autenticação baseada em JWT para garantir que apenas os usuários autenticados possam acessar suas notas.

1.2. Entregáveis

- **Interface de Usuário:** Design responsivo para criação, edição e visualização de notas.
- **Autenticação:** Implementação de login e registro com JWT.
- **Funcionalidade de Notas:** Recursos para criar, editar, excluir e visualizar notas.
- **Armazenamento Seguro:** Persistência de notas em um banco de dados seguro.
- **Documentação:** Manual do usuário e documentação técnica.

1.3. Exclusões

- **Sincronização em Tempo Real:** A sincronização em tempo real entre dispositivos não está incluída na versão inicial.
- **Exportação de Notas:** Recursos para exportar notas para diferentes formatos não estão incluídos.

---

2. Objetivos SMART

- **Específico:** Desenvolver uma aplicação de bloco de notas com autenticação JWT.
- **Mensurável:** A aplicação deve permitir criar, editar, e excluir notas, com um máximo de 99% de precisão nas operações CRUD e 100% de proteção contra acesso não autorizado.
- **Atingível:** Utilizando frameworks e bibliotecas modernas, como React para front-end e Node.js para back-end.
- **Relevante:** Atender às necessidades de usuários que desejam uma forma segura e prática de gerenciar suas notas.
- **Temporal:** Concluir o desenvolvimento da aplicação em 12 semanas.

---

3. Cronograma - Diagrama de Gantt

3.1. Planejamento

- **Definição de Requisitos:** 1 semana
- **Pesquisa e Análise de Tecnologia:** 1 semana

3.2. Design

- **Criação de Protótipos:** 2 semanas
- **Revisão e Aprovação do Design:** 1 semana

3.3. Desenvolvimento

- **Configuração do Ambiente e Banco de Dados:** 1 semana
- **Implementação do Back-end (Node.js, JWT, Banco de Dados):** 4 semanas
- **Desenvolvimento do Front-end (React, Interface de Usuário):** 4 semanas
- **Integração do Front-end e Back-end:** 2 semanas

3.4. Testes

- **Testes de Unidade:** 1 semana
- **Testes de Integração:** 1 semana
- **Testes de Aceitação:** 1 semana

3.5. Documentação e Lançamento

- **Criação de Documentação do Usuário e Técnica:** 1 semana
- **Preparação para Lançamento:** 1 semana
- **Lançamento:** 1 semana

3.6. Pós-Lançamento

- **Monitoramento e Suporte Inicial:** 2 semanas

---

4. Análise de Risco

4.1. Risco de Segurança

- **Descrição:** Acesso não autorizado às notas dos usuários.
- **Mitigação:** Implementar autenticação JWT e criptografia de dados.

4.2. Risco de Falhas de Software

- **Descrição:** Bugs e falhas na aplicação.
- **Mitigação:** Realizar testes rigorosos e revisão de código.

4.3. Risco de Desempenho

- **Descrição:** Lento tempo de resposta ou problemas de escalabilidade.
- **Mitigação:** Otimizar o código e realizar testes de carga.

4.4. Risco de Alterações no Requisito

- **Descrição:** Mudanças nos requisitos durante o desenvolvimento.

- **Mitigação:** Manter comunicação constante com as partes interessadas e gerenciar mudanças de forma eficaz.

4.5. Risco de Atrasos

- **Descrição:** Atrasos no cronograma.
- **Mitigação:** Estabelecer marcos intermediários e manter uma gestão de tempo eficaz.

---

5. Recursos

5.1. Recursos Humanos

- **Desenvolvedores:** 2 desenvolvedores (1 para back-end, 1 para front-end)
- **Designer:** 1 designer para UI/UX
- **QA:** 1 especialista em garantia de qualidade

5.2. Recursos Tecnológicos

- **Ferramentas de Desenvolvimento:** Visual Studio Code, GitHub
- **Frameworks e Bibliotecas:** React, Node.js, Express, JWT
- **Banco de Dados:** MongoDB ou MySQL
- **Infraestrutura:** Servidor para hospedagem da aplicação

5.3. Recursos Financeiros

- **Orçamento para Ferramentas e Licenças:** $2.000
- **Custos de Hospedagem e Domínio:** $500

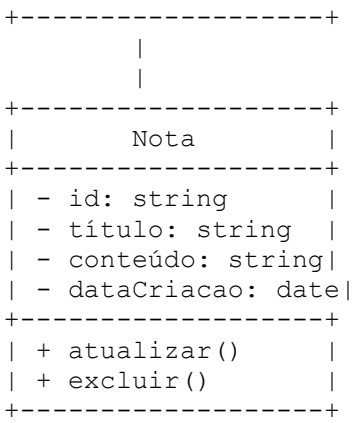
---

6. Diagramas

6.1. Diagrama de Classe

- **Descrição:** Representação das classes principais (Usuário, Nota) e suas relações.
- **Exemplo:**

```
plaintext
Copiar código
+--------------------+
|      Usuário       |
+--------------------+
| - id: string       |
| - email: string    |
| - senha: string    |
+--------------------+
| + criarNota()      |
| + editarNota()     |
| + deletarNota()    |
```



6.2. Diagrama de Caso de Uso

- **Descrição:** Mostra como os usuários interagem com a aplicação.
- **Exemplo:**

```
plaintext
Copiar código
[Usuário] -- (Login)
[Usuário] -- (Criar Nota)
[Usuário] -- (Editar Nota)
[Usuário] -- (Excluir Nota)
[Usuário] -- (Visualizar Nota)
```

6.3. Diagrama de Fluxo

- **Descrição:** Representa o fluxo de processos na aplicação.
- **Exemplo:**

```
plaintext
Copiar código
Início
  |
  v
[Tela de Login]
  |
  v
[Validação do Usuário] -- Se inválido --> [Mensagem de Erro]
  |
  v
[Tela Principal]
  |
  v
[Criar Nota] -- (Salvar Nota) --> [Banco de Dados]
  |
  v
[Editar Nota] -- (Atualizar Nota) --> [Banco de Dados]
  |
  v
[Excluir Nota] -- (Remover Nota) --> [Banco de Dados]
  |
  v
[Visualizar Nota]
  |
  V Fim
```