

## ROTEIRO BASE PARA ELABORAÇÃO DO TRABALHO STARTUP MODEL INTERMEDIATE

### PRIMEIRO BIMESTRE

1. Definição da força de trabalho (1 a 3 alunos)
  - a. Nomes: \_\_\_\_\_
2. Criação de empresa fictícia de TI.
  - a. Conhecer o sistema de criação de microempreendedor (MEI).  
<https://www.gov.br/empresas-e-negocios/pt-br/empreendedor>
    - i. Escolher CNAES relacionadas com a atividade desejada.
    - ii. Verificar taxas, obrigações e benefícios.
    - iii. Opcional efetuar realmente a criação da empresa (responsabilidade legal).
    - iv. Escolher nome fantasia, visão, missão, valores, logo.
3. Avaliação de problema proposto.
  - a. Escolha de problema a ser solucionado.
  - b. Avaliação de proposta de solução a ser entregue.
  - c. Definição das etapas do desenvolvimento.
    - i. Quem será responsável por cada atividade (requisitos, modelagem, interface, documentação, apresentação).
  - d. Elaboração de cronograma base para o projeto.

CRONOGRAMA DO PROJETO																	
ATIVIDADES	FEV			MAR			ABR				MAI			JUN/JUL			
Definição da empresa																	
Problema proposto																	
Coleta de requisitos																	
Documentação de requisitos																	
Apresentação								X								X	

- e. Busca por material de apoio (citar referências no trabalho seguindo normas ABNT).
4. Método de coleta de requisitos.
  - a. Realização de entrevista com roteiro planejado (presencial ou remota).
  - b. Elaboração de pesquisa para coleta de requisitos (online ou física).
  - c. Organização dos requisitos em funcionais ou não (definir prioridades).
  - d. Alinhamento dos requisitos em cronograma pode ajudar na segunda etapa.
5. Apresentação do documento de requisitos.
  - a. Entrega da versão mais atual do documento de requisitos.
  - b. Utilização de slides para apresentação (opcional – modelo disponível na blackboard).
  - c. Apresentação estilo PITCH de até 5 minutos.

## SEGUNDO BIMESTRE

6. Etapas do processo.
  - a. Uso de conhecimentos adquiridos na disciplina de modelagem de banco de dados.
  - b. Elaborar diagramas (DER) para indicar a estrutura de organização de dados.
  - c. Adicionar diagramas à documentação do projeto.
7. Design de layout e implementação.
  - a. Criação de layout para criação de protótipo das telas.
  - b. Criar páginas para cada tela do protótipo.
    - i. Interligar as telas com links simulando navegação.
    - ii. Utilizar recursos estáticos de HTML e CSS no processo.
  - c. Incluir prints e código na documentação.
8. Conferência de requisitos simulados.
  - a. Testar o funcionamento do protótipo.
  - b. Verificar se atende cada requisito funcional.
  - c. Realizar ajustes e melhorias se necessário.
9. Atualização da documentação do trabalho.
  - a. Ajustar o conteúdo do trabalho entregue no 1º bimestre.
  - b. Organizar o conteúdo adicional do 2º bimestre.
  - c. Telas e código podem ser incluídos como anexos.
  - d. Não esquecer referências bibliográficas se necessário.
  - e. Citar o uso de PROMPTS de ia se for o caso.
10. Apresentação do protótipo desenvolvido.
  - a. Entrega da versão mais atual do documento do projeto.
    - i. Apresentação dos membros da equipe.
  - b. Utilização de slides para apresentação (opcional – modelo disponível na blackboard).
  - c. Apresentação estilo PITCH de até 5 minutos.

## Anexo A – Modelo de Trabalho

### TÍTULO DO TRABALHO.

Subtítulo do trabalho.

Nome<sup>1</sup>

### RESUMO

Neste estudo, uma pesquisa bibliográfica foi realizada com o objetivo de ... (um parágrafo, sem recuo, com 150 a 250 palavras, espaçamento simples, justificado, tamanho 12 contendo objetivo, breve tema, breve metodologia e breve conclusão).

**Palavras-chave:** De 3 a 6 palavras mais importantes, frequentemente do resumo, separadas por ponto final e 1ª letra maiúscula, tamanho 12, sem negrito. Expressão ‘Palavras-chave’ tamanho 14, negrito, à esquerda.

### *ABSTRACT*

Tradução direta do resumo, seguindo as mesmas orientações de formatação.

**Keywords:** tradução direta das palavras-chave, seguindo as mesmas orientações.

### 1 Introdução

Contextualização do tema a ser discutido, enfatizando relevância, objetivo, metodologia e resumo das partes desenvolvidas. Fonte tamanho 12, espaçamento duplo, após o título Introdução deixar uma linha em branco. De um a dois parágrafos.

---

<sup>1</sup> E-mail: emaildoaluno@email.com

**2 Desenvolvimento** (negrito, tamanho 12, à esquerda, após o título Desenvolvimento deixar uma linha em branco l).

Subtítulo (Título segundo nível) – Item não obrigatório (tamanho 12, à esquerda, sem negrito, após o título Subtítulo deixar uma linha em branco).

Texto resultante da pesquisa bibliográfica com citações referenciadas ao final do texto. Quantidade de páginas varia de acordo com o tipo de trabalho científico a ser realizado.

Para este modelo, indica-se que o total de páginas do trabalho esteja entre 9 a 12 páginas, representando um artigo curto (paper).

2. 1 Subtítulo (Título segundo nível) – **Item não obrigatório**  
(tamanho 12, à esquerda, sem negrito, após o título Subtítulo deixar uma linha em branco)

Continuação do desenvolvimento...

### 3 Considerações Finais

Descrever como os objetivos foram atendidos e fique atento, pois o ideal é que esse tópico tenha no máximo 2 (dois) parágrafos e sem citação.

### 4 Referências Bibliográficas

Nome(s) do(s) autor(es), título, subtítulo (se houver), edição (se houver), local, editora e data de publicação.

Nome(s) do(s) autor(es) (se houver), título da matéria, nome do site, ano, endereço eletrônico e a data em que fizemos o acesso.

Dweck, C. S. (2017). **Mindset**: A nova psicologia do sucesso. Rio de Janeiro: Objetiva.

Elrod, H. (2016). **O milagre da manhã**: o segredo para transformar sua vida (antes das 8 horas). 1. ed. Rio de Janeiro: Best Seller, 2016.

Tavares, W. M. L. 01). **A indústria eletrônica no Brasil e seu impacto sobre a balança comercial**. Disponível em

<http://apache.camara.gov.br/portal/arquivos/Camara/internet/publicacoes/estnottec/pdf/108604.pdf>. Acessado em 10 de janeiro de 2020.

## Anexo B – Modelo de documentação de software

### Documentação do Software [Nome do Software][Versão]

#### 1. Introdução

##### 1.1. Objetivo

Este documento tem como finalidade descrever detalhadamente o software desenvolvido, incluindo requisitos, estrutura do código-fonte, testes realizados e demais informações relevantes.

##### 1.2. Escopo

O software foi desenvolvido para [descrever a finalidade do software] e será utilizado por [descrever o público-alvo].

##### 1.3. Definições, Acrônimos e Abreviações

- **RF:** Requisito Funcional
- **RNF:** Requisito Não Funcional
- **API:** Interface de Programação de Aplicação
- **CI/CD:** Integração Contínua e Entrega Contínua

##### 1.4. Referências

[Listar normas, manuais ou referências técnicas utilizadas no projeto]

##### Sites\*:

SOBRENOME, Nome. **Título da matéria.** Nome do site, ano. Disponível em: . Acesso em: dia, mês e ano.

##### Livro online\*:

AUTOR OU ORGANIZAÇÃO. **Título.** Cidade: Responsável pela publicação, ano. Total de páginas. Disponível em: URL. Acesso em: dia, mês e ano.

##### Livro físico:

Nome do autor. Título do livro: Subtítulo (se houver). Edição (se houver). Local de publicação: Editora, Ano de publicação.

\*Caso não tenha data, indicar s.d.

## 2. Requisitos do Software

### 2.1. Requisitos Funcionais

ID	Descrição	Prioridade
RF001	O sistema deve permitir o cadastro de usuários.	Alta
RF002	O sistema deve permitir login e autenticação de usuários.	Alta
RF003	O sistema deve permitir a geração de relatórios.	Média

### 2.2. Requisitos Não Funcionais

ID	Descrição	Prioridade
RNF001	O sistema deve responder às solicitações em menos de 2 segundos.	Alta
RNF002	O sistema deve estar disponível 99,9% do tempo.	Alta
RNF003	O sistema deve utilizar criptografia para armazenamento de senhas.	Alta

## 3. Arquitetura e Código-Fonte

### 3.1. Tecnologias Utilizadas

- Linguagem de programação: [Ex: Python, Java, JavaScript]
- Banco de dados: [Ex: MySQL, PostgreSQL]
- Frameworks: [Ex: Django, Spring Boot, React]
- Controle de versão: [Ex: GitHub, GitLab]

### 3.2. Estrutura do Código-Fonte

```
/project
├── src/           # Código-fonte principal
├── tests/         # Testes automatizados
├── docs/          # Documentação do projeto
├── config/        # Configurações do sistema
├── README.md      # Documentação inicial
└── requirements.txt # Dependências do projeto
```

### 3.3. Principais Componentes

- **Módulo de Autenticação:** Gerencia usuários e permissões.
- **Módulo de Relatórios:** Gera relatórios personalizados para os usuários.
- **Módulo de API:** Responsável por comunicação entre serviços.

## 4. Testes Realizados

### 4.1. Tipos de Testes

- **Testes Unitários:** Validam funcionalidades individuais do sistema.
- **Testes de Integração:** Verificam a comunicação entre módulos.
- **Testes de Aceitação:** Avaliam se o software atende aos requisitos do cliente.

## 4.2. Ferramentas Utilizadas

- [Ex: JUnit, Selenium, Postman]

## 4.3. Cobertura de Testes

- Cobertura de código: [Ex: 85%]
- Número de testes executados: [Ex: 150]
- Erros encontrados e corrigidos: [Descrever brevemente]

## 5. Implantação e Manutenção

### 5.1. Processo de Deploy

- Ambiente de produção: [Ex: AWS, Azure, Heroku]
- Pipeline CI/CD configurado: [Sim/Não]
- Backup e recuperação de dados: [Descrever estratégia]

### 5.2. Monitoramento e Suporte

- Ferramentas de monitoramento: [Ex: Prometheus, Grafana]
- Política de manutenção: [Ex: Atualizações trimestrais]

## 6. Conclusão

Este documento serve como guia para desenvolvimento, manutenção e evolução do software, garantindo sua qualidade e continuidade. Para mais detalhes, consulte a documentação técnica e código-fonte disponível no repositório do projeto.

## 7. Anexos

[Listar diagramas, fluxogramas e outros documentos complementares]

### Fotos do produto funcionando (teste de protótipo)

[]

Figura 1: protótipo montado.

[]

Figura 2: protótipo em funcionamento.