

Documento do Projeto: Leitura Descomplicada (Versão Revisada e Comentada)

Olá! Analisei o texto que você enviou (Documento.txt) e comparei com a estrutura detalhada que definimos (revised_structure.md). O início está muito bom e bem alinhado!

Abaixo, apresento o seu texto com algumas correções, comentários e, principalmente, **placeholders indicando as seções que ainda precisam ser desenvolvidas ou completadas**, seguindo a estrutura completa que planejamos. Os comentários estão marcados com <!-- COMENTÁRIO: ... --> ou [PLACEHOLDER: ...] , e as sugestões de visuais estão mantidas como [INSERIR DIAGRAMA/IMAGEM: ...] . Lembre-se que esses visuais são essenciais para atender à sua preferência e às boas práticas.

[PLACEHOLDER: Inserir Capa aqui] * **Conteúdo:** Nome do Projeto ("Leitura Descomplicada"), Seu Nome Completo e RA, Nome da Disciplina, Instituição, Mês/Ano. * **Visual:** Logo da instituição (se aplicável), talvez um elemento visual sutil relacionado ao tema do projeto.

1. Introdução

1.1. Visão Geral e Contexto

O projeto "Leitura Descomplicada" surge como uma solução inovadora para um problema contemporâneo: a dificuldade de jovens e adultos em manter uma rotina de leitura consistente em meio à crescente distração proporcionada por mídias digitais rápidas e efêmeras. Em um cenário onde o tempo de atenção é fragmentado e a busca por gratificação instantânea prevalece, a leitura, um hábito fundamental para o desenvolvimento crítico e intelectual, muitas vezes é relegada a segundo plano. Conforme discutido em [Referência à aula_doc_parte1.txt ou aula específica], a capacidade de concentração e a busca por informações rápidas impactam diretamente o engajamento com conteúdos mais longos.

Este projeto visa resgatar o prazer da leitura e torná-la mais acessível e engajadora, especialmente para o público jovem, que está constantemente imerso em conteúdos de

curta duração. Através de uma plataforma web interativa, buscamos criar um ambiente que não apenas recomende livros, mas também envolva o usuário de maneira dinâmica, incentivando a descoberta e a formação de hábitos de leitura.

[INSERIR DIAGRAMA: Diagrama de Contexto (Nível 0) mostrando "Leitura Descomplicada" no centro e as interações com atores externos (Usuário, talvez Administrador se houver). Ajuda a situar o projeto rapidamente.] [INSERIR IMAGEM: Opcional: Screenshot da Tela Inicial: Uma imagem vale mais que mil palavras para dar a primeira impressão.]

1.2. Objetivos (SMART)

Os objetivos do projeto "Leitura Descomplicada" são definidos para guiar o desenvolvimento e mensurar o sucesso da iniciativa. Eles seguem o padrão SMART (Específico, Mensurável, Atingível, Relevante, Temporal), conforme abordado em [Referência à aula_doc_parte1.txt ou aula específica]:

- **Objetivo Principal:**

- **Incentivar a Leitura:** Proporcionar uma ferramenta interativa que motive o público jovem a incorporar o hábito da leitura em suas rotinas diárias, oferecendo recomendações personalizadas e atividades lúdicas.

- **Objetivos Secundários:**

- **Técnico (SMART):**

- **Específico:** Desenvolver uma aplicação web full-stack utilizando Node.js para o backend, MySQL para o banco de dados, e HTML/CSS/JavaScript puro para o frontend.
- **Mensurável:** Implementar as funcionalidades de cadastro/login, quiz interativo, carrossel de recomendações e dashboard de acompanhamento de leitura até o final do sprint [Mês/Ano].
- **Atingível:** A meta é atingível considerando as tecnologias estudadas em sala de aula e os recursos disponíveis.
- **Relevante:** A aplicação destas tecnologias é diretamente relevante para o desenvolvimento das habilidades de um profissional de TI.
- **Temporal:** Concluir o desenvolvimento e a implantação da versão inicial da plataforma até [Data específica de entrega].

- **De Negócio/Usuário (SMART):**

- **Específico:** Criar um quiz que, após a interação, recomende pelo menos [X] títulos de livros alinhados aos interesses do usuário.
- **Mensurável:** Obter uma taxa de satisfação de [Y]% dos usuários com as recomendações de livros, medida através de um formulário de feedback pós-uso.

- **Atingível:** Com base na lógica de filtragem implementada, as recomendações serão relevantes e o feedback demonstra o alinhamento com as expectativas do usuário.
- **Relevante:** Melhorar a experiência do usuário e a eficácia das recomendações para fomentar a leitura.
- **Temporal:** Coletar feedback e analisar a taxa de satisfação nos primeiros [Z] meses de uso.

1.3. Justificativa

A leitura é uma ferramenta insubstituível para o desenvolvimento cognitivo, a ampliação do vocabulário, o estímulo à criatividade e o aprimoramento do senso crítico. No entanto, o avanço das redes sociais e do conteúdo rápido tem desviado a atenção, especialmente das novas gerações, da profundidade que a leitura proporciona. O projeto "Leitura Descomplicada" se justifica pela urgência de oferecer uma alternativa atraente e eficaz para reverter essa tendência. Ao tornar a leitura mais acessível e adaptada ao estilo de vida digital, a plataforma não apenas preenche uma lacuna no mercado de aplicativos de incentivo à leitura, mas também contribui para a formação de indivíduos mais informados e engajados. A relevância pessoal neste projeto reside na crença de que a leitura é um pilar fundamental para o crescimento individual e social, e desenvolver uma ferramenta que facilite esse acesso é uma contribuição significativa.

1.4. Público-Alvo

O público-alvo principal do "Leitura Descomplicada" são **jovens e adultos entre [IDADE MÍNIMA] e [IDADE MÁXIMA] anos**, que possuem acesso à internet e dispositivos digitais (smartphones, tablets, computadores). As características desse público incluem:

- **Hábitos:** Consumidores ávidos de conteúdo digital, acostumados com interfaces intuitivas e interativas, porém com dificuldade em focar em leituras longas.
- **Necessidades:** Buscam recomendações de entretenimento e aprendizado, mas muitas vezes não sabem por onde começar ou sentem-se sobrecarregados pela quantidade de opções.
- **Dores:** Frustração por não conseguir manter o hábito de leitura, falta de tempo percebida, dificuldade em encontrar livros que realmente os interessem e distração fácil.

[INSERIR DIAGRAMA: Diagrama de Persona Simplificado (Crie 1 ou 2 personas representando seus usuários típicos, com nome fictício, foto (pode ser genérica/ícone), características, objetivos e frustrações relacionadas à leitura. Torna o público mais concreto.)]

1.5. Escopo do Projeto

O escopo do projeto "Leitura Descomplicada" delimita CLARAMENTE o que está incluído e o que está excluído nesta fase inicial, conforme as orientações.

Inclusões (Funcionalidades Principais):

- **Módulo de Autenticação:**
 - Cadastro de novos usuários.
 - Login de usuários existentes.
- **Quiz Interativo de Preferências:**
 - Formulário com perguntas sobre gêneros, autores, e temas de interesse.
 - Lógica para processar as respostas e gerar recomendações de livros.
- **Carrossel de Recomendações:**
 - Exibição dinâmica de capas de livros e títulos recomendados.
 - Possibilidade de navegar entre as recomendações.
- **Formulário de Feedback:**
 - Envio de feedback sobre a plataforma ou sugestões de livros.
- **Dashboard do Usuário:**
 - Visualização de dados sobre o progresso de leitura (ex: livros lidos, tempo de leitura).
 - Status de quiz (se completou ou não).
- **Backend (API Node.js):**
 - Recebimento e processamento de requisições do frontend.
 - Lógica de negócios e comunicação com o banco de dados.
 - Endpoints para usuários, livros, quizzes, etc.
- **Banco de Dados (MySQL):**
 - Armazenamento de dados de usuários, livros, quiz, feedback, etc.
 - Hospedado em uma Máquina Virtual (VM) Linux.
- **Infraestrutura:**
 - Configuração de uma VM Linux para o banco de dados.

Exclusões (Funcionalidades Não Implementadas Nesta Fase):

- Integração com APIs de terceiros (ex: Goodreads, Amazon para detalhes de livros).
- Sistema de leitura de livros na própria plataforma.
- Funcionalidades sociais (compartilhamento, comentários, seguir usuários).
- Gamificação avançada (pontuações, conquistas).
- Notificações push.
- Recuperação de senha via e-mail.
- Relatórios complexos para o administrador.

[INSERIR IMAGEM: Opcional: Tabela de Inclusões/Exclusões (se quiser formatar visualmente diferente das listas). Pode referenciar novamente o Diagrama de Contexto aqui.]

1.6. Premissas e Restrições

O desenvolvimento do projeto "Leitura Descomplicada" é baseado nas seguintes premissas e está sujeito às seguintes restrições, conforme abordado em [Referência à aula_doc_parte2.txt ou aula específica]:

Premissas:

- Disponibilidade de acesso à internet para o usuário final.
- Acesso e funcionamento adequado da máquina virtual (VM) Linux para o banco de dados.
- Conhecimento básico de HTML, CSS, JavaScript, Node.js e MySQL por parte da equipe de desenvolvimento (neste caso, o desenvolvedor individual).
- As bibliotecas e frameworks utilizados terão documentação acessível e funcional.
- [Outras premissas específicas do seu projeto, ex: dados de livros serão providos manually.]

Restrições:

- **Prazo:** O projeto deve ser entregue até [Data limite do projeto/semestre].
 - **Orçamento:** Orçamento zero para ferramentas pagas ou serviços de hospedagem (utilização de ferramentas gratuitas ou de código aberto).
 - **Tecnologias Obrigatórias:** A stack tecnológica é restrita a Node.js (Express), MySQL e HTML/CSS/JavaScript puro no frontend, conforme requisitos do curso.
 - **Recursos Humanos:** Projeto individual, o que limita a capacidade de desenvolvimento e testes simultâneos.
 - **Ambiente de Desenvolvimento:** Utilização de ambiente local para desenvolvimento e testes, com a VM para simulação de um ambiente de produção para o BD.
-

2. Planejamento e Metodologia

2.1. Metodologia de Desenvolvimento

Para o desenvolvimento do "Leitura Descomplicada", foi adotada uma metodologia **Scrum adaptada ao contexto de um projeto individual**. Embora os rituais completos de um Scrum team não sejam aplicáveis, a essência foi mantida:

- **Backlog de Produto:** Uma lista de todas as funcionalidades e melhorias desejadas, priorizadas.
- **Sprints/Iterações:** O desenvolvimento foi dividido em ciclos curtos e iterativos (mesmo que informais), focando na entrega de funcionalidades incrementais e funcionais ao final de cada período.
- **Entregas Contínuas:** Priorização de ter uma versão funcional do sistema a cada etapa, permitindo testes e refinamento.
- **Flexibilidade:** Capacidade de adaptar e repriorizar tarefas conforme o aprendizado e os desafios surgiam.

Essa escolha se justifica pela **flexibilidade** que permite o gerenciamento eficiente do tempo e a **familiaridade** com o conceito de entregas incrementais, algo crucial para projetos acadêmicos com prazos definidos. Demonstra um conhecimento prático sobre processos de desenvolvimento ágil, mesmo em um contexto solo.

2.2. Cronograma e Marcos (Milestones)

O cronograma macro do projeto seguiu as seguintes fases e marcos, que são pontos de entrega cruciais para o projeto, conforme discutido em [Referência à aula_doc_parte2.txt ou aula específica]:

Marco Principal	Data Prevista/Realizada
Setup Inicial	[Data]
(Ambiente Dev, Git, Trello)	
Frontend Básico	[Data]
(Estrutura HTML/CSS das telas principais)	
Backend Inicial	[Data]
(API e Conexão BD)	

Marco Principal	Data Prevista/Realizada
Módulo de Autenticação	[Data]
(Cadastro/Login)	
Quiz Funcional	[Data]
(Lógica e Persistência)	
Dashboard e Recomendações	[Data]
(Exibição de dados)	
Testes e Ajustes Finais	[Data]
Apresentação Final	[Data]

[INSERIR IMAGEM: Opcional: Gráfico de Gantt Simplificado (Pode ser feito em ferramentas online ou até uma tabela bem formatada. Mostra a linha do tempo visualmente.)]

2.3. Ferramentas Utilizadas

A tabela a seguir lista todas as ferramentas utilizadas no desenvolvimento do projeto, com uma breve justificativa para cada escolha:

Ferramenta	Propósito Principal	Justificativa no Projeto
VSCode	Editor de código	Ambiente de desenvolvimento leve, robusto e com vasta gama de extensões para JS/Node/SQL.
Git/GitHub	Controle de versão e colaboração (repositório)	Essencial para versionamento do código, histórico de alterações e organização do projeto.
Trello	Gerenciamento de tarefas e backlog	Ferramenta ágil e visual para organizar as atividades e o fluxo de trabalho individual.
Figma/Canva	Prototipagem e Design Visual	Criação de layouts, mockups e elementos visuais para o frontend.

Ferramenta	Propósito Principal	Justificativa no Projeto
MySQL Workbench	Gerenciamento e modelagem de banco de dados MySQL	Ferramenta gráfica para projetar, administrar e interagir com o banco de dados.
VirtualBox	Virtualização de ambiente Linux (VM)	Criação e execução da VM Linux para hospedar o banco de dados MySQL.
Node.js	Ambiente de execução JavaScript (Backend)	Escolhido por ser uma tecnologia moderna, eficiente e requisito do curso.
Express.js	Framework web para Node.js (Backend)	Facilita a criação de APIs RESTful de forma rápida e organizada.
MySQL	Sistema de Gerenciamento de Banco de Dados Relacional	Banco de dados robusto, amplamente utilizado e requisito do curso.
HTML5/CSS3/JS	Desenvolvimento Frontend	Tecnologias padrão da web para construção de interfaces de usuário.
[Outra Ferramenta, se usou]	[Propósito]	[Justificativa]

2.4. Gerenciamento de Projeto (Trello)

O Trello foi a ferramenta central para o gerenciamento individual das tarefas do projeto. Um quadro específico foi criado, contendo as seguintes listas principais, representando as fases de um fluxo de trabalho ágil:

- **Backlog:** Todas as funcionalidades e melhorias desejadas, priorizadas, aguardando o momento de serem desenvolvidas.
- **A Fazer (To Do):** Tarefas que foram selecionadas do backlog e estão prontas para serem iniciadas.
- **Em Andamento (Doing):** Tarefas que estão atualmente em desenvolvimento.
- **Concluído (Done):** Tarefas finalizadas e verificadas, demonstrando o progresso.

Cada cartão no Trello representava uma funcionalidade ou uma tarefa específica, com descrição, checklists (para sub-tarefas), prazos e anexos quando necessário. Essa

organização visual permitiu um acompanhamento claro do progresso e uma gestão eficaz do tempo, mesmo em um projeto de um único desenvolvedor.

[INSERIR IMAGEM: Screenshot do Quadro Trello do Projeto (Mostre a organização na prática, com as listas To Do, Doing, Done e alguns cartões de exemplo preenchidos.)]

2.5. Controle de Versão (Git/GitHub)

O controle de versão foi realizado utilizando Git, com o repositório hospedado no GitHub. A estratégia de branches adotada foi simplificada e eficiente para um projeto individual:

- **main (ou master)** : Branch principal, contendo apenas o código estável e funcional, representando as versões liberadas do projeto.
- **feature/[nome-da-funcionalidade]** : Branches criadas para o desenvolvimento de funcionalidades específicas, isolando o trabalho e permitindo experimentação sem afetar a estabilidade do código principal. Ao final da funcionalidade, a feature branch era mergeada na **main**.

A frequência de commits foi constante, garantindo que o histórico de alterações fosse granular e que cada alteração significativa fosse registrada. As mensagens de commit seguiram uma convenção básica para clareza e rastreabilidade.

[INSERIR TEXTO/IMAGEM: Exemplo de Mensagem de Commit (Cole um exemplo real ou um screenshot do log do git).] [INSERIR DIAGRAMA: Opcional: Diagrama Simples de Fluxo de Branch (Se usou mais que main/feature, mostre visualmente).] **Link para o Repositório:** [Link]

[PLACEHOLDER: SEÇÃO 3 - REQUISITOS DO SISTEMA]

[PLACEHOLDER: SEÇÃO 4 - ARQUITETURA DO SISTEMA]

[PLACEHOLDER: SEÇÃO 5 - MODELAGEM DE DADOS]

[PLACEHOLDER: SEÇÃO 6 - IMPLEMENTAÇÃO FRONTEND]

[PLACEHOLDER: SEÇÃO 7 - IMPLEMENTAÇÃO BACKEND (JavaScript/Node.js)]

[PLACEHOLDER: SEÇÃO 8 - ASPECTOS SOCIOEMOCIONAIS]

[PLACEHOLDER: SEÇÃO 9 - TESTES]

[PLACEHOLDER: SEÇÃO 10 - CONCLUSÃO]

[PLACEHOLDER: SEÇÃO 11 - REFERÊNCIAS BIBLIOGRÁFICAS]
