Curso Superior de Desenvolvimento de Software Multiplataforma

Projeto Interdisciplinar

Banco de Dados Não Relacional

Desenvolvimento Web III

Gestão Ágil de Projetos de Software

Interação Humano Computador

Técnicas de Programação II

Orientadores

Profº Jones Artur Gonçalves

Profª Maria Janaína da Silva Ferreira

Profº Ricardo Roberto Leme

Profº Rodrigo De Paula Diver

Votorantim

Junho, 2024

1. Sumário

Observação:

Apresentação do Projeto no GitHub

A gestão de código e documentação em um repositório centralizado é uma prática essencial em projetos de desenvolvimento de software. GitHub é uma das plataformas mais utilizadas para este propósito, facilitando a colaboração, controle de versão, e revisão de código. Nesta etapa final, vocês irão consolidar todo o trabalho realizado ao longo do projeto em um repositório no GitHub.

Cada equipe deverá criar um repositório no GitHub para o projeto de desenvolvimento de software. Este repositório deve incluir todos os elementos desenvolvidos no projeto, como o backlog do produto, user stories, planejamento das sprints, e demonstração do produto. A documentação deve ser clara, organizada e refletir todo o trabalho realizado ao longo do ciclo de desenvolvimento.

Formato da Entrega:

Repositório GitHub: O repositório deve estar completo e acessível publicamente até a data de entrega.

Documentação Completa: Todas as documentações, incluindo backlog, user stories, planejamento das sprints, revisões de sprint, código-fonte, demonstrações, tecnologias utilizadas, e reflexões, devem estar incluídas e organizadas.

Apresentação Final: Prepare uma breve apresentação para demonstrar o repositório GitHub, destacando como cada parte do trabalho foi organizada e documentada.

1. Descrição do Projeto

1.1. Sobre o Projeto de Software (Objetivo)

Nesse tópico:

- Definir, de forma resumida, o que será o software.
- Descrever seus principais objetivos.
- Apontar qual é o problema a ser solucionado
- Apontar qual é o público-alvo

Se houver restrições, coloque-as nesse tópico.

1.2. Justificativa

Nesse tópico, escreva sobre como surgiu a ideia de desenvolvimento do site e quais problemas serão resolvidos com ele.

Pense na pergunta: Por que desenvolver esse site?

1.3. Logomarca

Insira a logomarca.

2. Entregas das Sprints

Ness tópico do projeto de desenvolvimento de software seguindo a metodologia Scrum, que é uma das abordagens mais populares no gerenciamento ágil de projetos. Cada equipe deverá preparar uma apresentação detalhada sobre as sprints desenvolvidas durante o projeto. Esta apresentação servirá tanto como uma entrega formal do trabalho realizado quanto como uma oportunidade para reflexão e aprendizado sobre o processo Scrum aplicado.

Explique as metas de cada sprint e os entregáveis específicos.

3. Apresentação Final

Neste tópico deve ser realizada uma Demonstração de Produto. Inclua vídeos ou capturas de tela do produto em funcionamento, se aplicável, para apoiar a demonstração prática.

4. Backlogs & User Stories

No desenvolvimento ágil de projetos usando Scrum, a definição e gerenciamento de backlog e user stories são elementos fundamentais. O backlog do produto é uma lista ordenada de tudo o que é necessário no produto, enquanto as user stories são descrições curtas e simples de uma funcionalidade contada da perspectiva do usuário final.

Para este exercício, vocês deverão apresentar o backlog e as user stories desenvolvidas para o seu projeto, explicando como cada item foi priorizado e detalhado ao longo do ciclo de desenvolvimento.

Cada equipe deverá preparar uma apresentação detalhada do backlog do produto e das user stories criadas durante o desenvolvimento do projeto. Esta apresentação deve demonstrar o entendimento da importância desses elementos no processo Scrum e a capacidade de criar descrições claras e úteis para o desenvolvimento ágil.

Objetivos Específicos:

Backlog do Produto:

Descrição Completa: Apresente o backlog do produto, incluindo todos os itens atualmente no backlog. Certifique-se de que a lista está ordenada por prioridade.

User Stories:

Criação de User Stories: Para cada item de maior prioridade no backlog, apresente as user stories correspondentes. Cada user story deve seguir o formato padrão (Como [tipo de usuário], eu quero [ação], para que [benefício]).

Critérios de Aceitação: Inclua critérios de aceitação claros para cada user story. Estes critérios ajudam a definir quando uma user story está completa e funcionando conforme esperado.

Ferramentas Utilizadas: Apresente as ferramentas utilizadas para gerenciar o backlog e as user stories (por exemplo, Trello, Jira, etc.). Mostre exemplos concretos do uso dessas ferramentas no projeto.

5. Protótipo

Prototipagem: Adicione documentos ou links para os protótipos desenvolvidos, descrevendo as ferramentas utilizadas para sua criação e explicando as funcionalidades principais demonstradas.

6. Documentação do Projeto

Documentação do Projeto: Inclua uma pasta /docs com todos os documentos relevantes, como o backlog, user stories, planejamento das sprints, revisões de sprint, lições aprendidas e reflexões.

Os Itens de 6.1 a 6.3devem ser inclusos dentro desta documentação.

6.1. Levantamento de Requisitos

Nesse tópico, é necessário descrever a técnica que foi utilizada para levantamento dos requisitos. Por exemplo:

- Questionários ou entrevistas com possíveis usuários
- Análise do sistema antigo da empresa (caso exista uma empresa para a qual o software foi desenvolvido)
- Pesquisa de mercado.

Se necessário, inclua documentos (coloque-os no Apêndice)

Também poderão ser descritas **ferramentas existentes no mercado** com funcionalidades semelhantes e que tenham sido utilizadas como base para a definição do projeto.

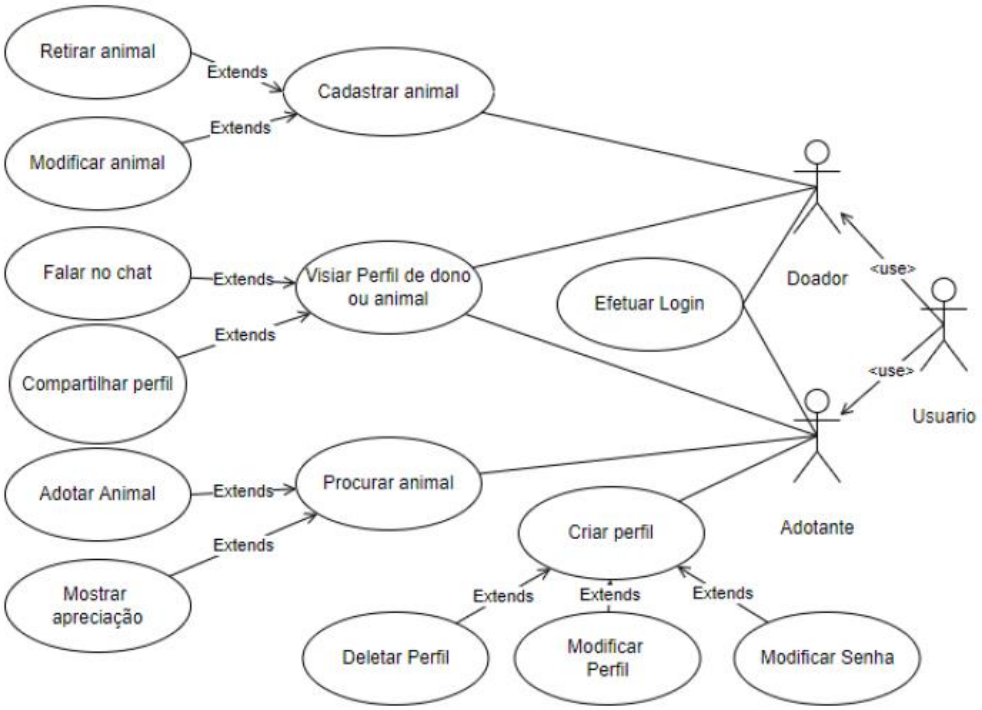
6.2. Requisitos Funcionais

Nesse tópico, é necessário descrever quais são os requisitos funcionais da aplicação a ser desenvolvida, ou seja, aqueles que definem as funções que o sistema deve oferecer.

6.3. Diagrama de Caso de Uso

Insira seu diagrama de Casos de Uso.

Exemplo:



Fonte: Autoria Própria

6.4. Requisitos Não Funcionais

Nesse tópico, é necessário descrever quais são os requisitos NÃO funcionais da aplicação a ser desenvolvida. Os requisitos não funcionais descrevem as características e as qualidades que o sistema deve possuir, além das funcionalidades. Eles estão relacionados ao desempenho, usabilidade, segurança, confiabilidade e outras propriedades do sistema.

Exemplos de requisitos não funcionais:

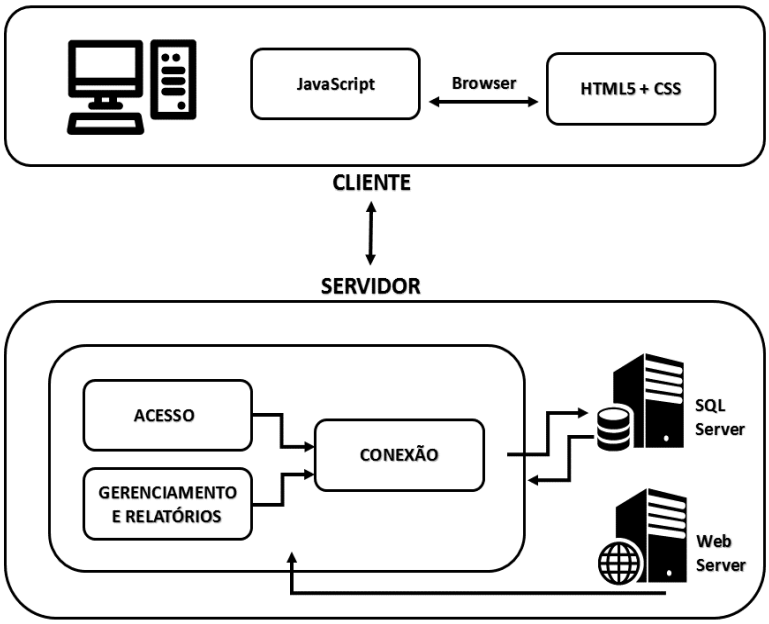
- tempo de resposta
- capacidade de processamento
- facilidade de uso
- proteção de dados
- confidencialidade
- tolerância a falhas.

7. PROJETO DO SOFTWARE

7.1. Arquitetura da Aplicação

Nesse tópico, apresentar de maneira sucinta qual foi o modelo arquitetural escolhido para o projeto. Além disso, é interessante incluir figuras facilitando o entendimento dos componentes, conforme exemplo apresentado na Figura 2.

Figura 1 - Arquitetura do Software



Fonte: Autoria própria

7.2. Tecnologias Utilizadas

Nesse tópico, é necessário descrever as tecnologias que serão utilizadas no desenvolvimento da aplicação, como por exemplo linguagem de programação, framework, banco de dados, API e hardware.

8. Integração e Papéis da Equipe

Equipe: Crie uma lista com todos os integrantes do grupo e seus papéis no projeto. Especifique quem atuou como Scrum Master, Product Owner e membros do time de desenvolvimento. Inclua uma breve descrição das responsabilidades e contribuições de cada membro.

9. Integração com Outras Disciplinas:

Os itens solicitados por cada uma das disciplinas a seguir devem ser incluídos no GitHub como parte do projeto.

Banco de Dados Não Relacional:

Os discentes terão a tarefa de criar um servidor backend RESTful usando o Node.js e integrá-lo com o MongoDB na nuvem. Recomenda-se que os discentes utilizem como base o projeto desenvolvido em aula como referência, mas eles tem a liberdade de estender e personalizar conforme necessário.
Requisitos do Projeto:

1. **Estrutura de Dados:** Criar uma Collection no MongoDB (livre escolha) que represente os dados de sua aplicação. Certifique-se de que essa Collection tenha no mínimo 5 atributos. (é obrigatório ter um campo do tipo data, int e valor com decimais). Note que não estamos considerando os campos padrão como _id, created_at, updated_at, etc.

2. **Controle de Versão:** O projeto deve ser hospedado de forma pública no GitHub. Certifique-se de que seu repositório esteja organizado e contenha um README.md que liste os integrantes do grupo e explique o propósito do projeto. Adicione também o link da API pública.

3. **API RESTful:** Desenvolva uma API RESTful completa que permita a realização das operações básicas: GET, POST, PUT e DELETE. Cada operação deve ser mapeada para as rotas apropriadas no seu servidor.

4. **Hospedagem:** Publique seu servidor backend em uma plataforma de hospedagem, como o Vercel ou uma ferramenta similar. Certifique-se de que a API esteja acessível publicamente.

5. **Documentação das Chamadas REST:** Utilize a extensão REST Client no Visual Studio Code para criar e apresentar as chamadas REST (GET, POST, PUT, DELETE) que interagem com os dados no banco de dados na nuvem. Documente essas chamadas de maneira clara no seu projeto.

6. **Consulta com Operadores:** Implemente pelo menos uma rota GET que utilize dois ou mais operadores de comparação, como $gt, $gte, $lt, $lte, $in, $nin, $ne, $nin, e um operador lógico, como $and, $or, $nor. Isso demonstrará sua compreensão de consultas complexas no MongoDB.

7. **Validação do conteúdo:** Utilize a biblioteca express-validator para efetuar todas as validações necessárias na collection, nos métodos POST e PUT

8. **Entrega:** Envie um link para o repositório do GitHub via a Tarefa do Microsoft Teams até a data combinada de entrega. Certifique-se de incluir o nome de todos os integrantes do grupo no README do projeto para facilitar a identificação dos colaboradores, assim como o link da API pública.
Lembre-se de que o projeto deve ser funcional, bem documentado e seguir as melhores práticas de desenvolvimento web com Node.js e MongoDB.

Segunda Entrega

Na segunda entrega do projeto, os discentes terão a tarefa de criar um sistema de autenticação JWT (JSON Web Tokens) e integrar uma interface de usuário (UI) com as APIs RESTful que foram desenvolvidas na entrega anterior. O objetivo é garantir que a UI seja capaz de consumir as APIs e exibir as mensagens geradas pela API de forma autenticada.
Requisitos do Projeto:

1. **Implementação do JWT:** Crie um sistema de autenticação JWT para proteger as rotas da sua API. Os usuários devem poder se autenticar, receber um token JWT válido e usá-lo para acessar as rotas protegidas.

2. **Criação da UI:** Desenvolva uma interface de usuário (pode ser uma página da web com HTML, CSS e Javascript ou um framework como Angular, React, Vue, etc.) que permita aos usuários se autenticarem e interagirem com as operações da API. (incluir, editar e apagar o registro)

3. **Integração com API:** A UI deve ser capaz de fazer solicitações às APIs para realizar operações básicas, incluindo GET, POST, PUT e DELETE. As solicitações devem incluir obrigatoriamente o token JWT para autenticação.

4. **Exibição de Mensagens:** Garanta que a UI exiba as mensagens geradas pela API de forma clara e legível para o usuário.

5. **Documentação da API com Swagger:** Adicione documentação Swagger (https://swagger-autogen.github.io/docs/) à sua API para facilitar a compreensão e o uso por parte dos desenvolvedores. Isso inclui a descrição detalhada de cada rota, os parâmetros necessários, os tipos de resposta esperados e exemplos de solicitações.

6. **Entrega:** Envie um link para o repositório do GitHub via a Tarefa do Microsoft Teams. Certifique-se de incluir o nome de todos os integrantes do

grupo no README do projeto e o link da UI para facilitar a identificação dos colaboradores, assim como o link da API pública, se houver. (Apenas um componente do grupo deve efetuar a entrega no Teams)

Técnicas de Programação II

Objetivos: Os discentes deverão aperfeiçoar as interfaces de administração de seus respectivos projetos integradores, adicionando novas funcionalidades e comportamentos. Haverá acesso ao banco de dados nessa etapa.

Primeira Entrega: Documento descrevendo as modificações e aperfeiçoamentos que serão implementados. Diagrama UML de classes, incluindo as classes de acesso ao banco de dados.
Prazo de entrega: 10ª semana.

Segunda Entrega: Descrição detalhada dos testes de validação, utilizando de forma mais profunda os conceitos de TDD.
Prazo de entrega: 12ª semana.

Terceira Entrega: Classes de acesso ao banco de dados e apresentação das novas funcionalidades, acompanhado dos testes de validação e modelos dos bancos de dados.
Prazo de entrega: 14ª semana.

Quarta Entrega: Integração das novas classes ao projeto, acompanhado dos testes de validação.
Prazo de entrega: 16ª semana.

Quinta Entrega: Apresentação do projeto funcional e entrega da documentação atualizada.
Prazo de entrega: 17ª semana.

Interação Humano Computador

Objetivos: O aluno deverá aperfeiçoar a interface desenvolvida, implementando os requisitos da IHC, trabalhando os conceitos de usabilidade e acessibilidade. A entrega final será o documento no formato pdf, com normas ABNT, contento os seguintes tópicos:

- Capa
- Equipe
- Nome do site
- Técnica para identificação dos usuários (questionário, braimstorm etc)
- Definição das personas
- Definição dos cenários
- Estrutura da informação (card sorting)
- Menus e navegação
- Protótipo de baixa fidelidade
- Definição dos símbolos
- Definição das cores
- Definição dos Padrões
- Protótipo de alta fidelidade
- Avaliação Heurística

Para cada tópico será necessário elencar os objetivos, a proposta e o resultado obtido.

Desenvolvimento Web III

Objetivos: O aluno deverá criar serviços e microserviços para a implementação do back end de sua aplicação.

A entrega será o projeto no formato .zip e o link da aplicação publicada no GitHub. Os requisitos são os definidos na disciplina Banco de Dados não relacional.

10. Exemplo de um projeto

Segue o link com um exemplo de um projeto criado por uma outra turma, que pode servir como um modelo para o projeto.

https://github.com/The-Bugger-Ducks/help-duck-documentation