FECAP

**PROJETO ESTUDA MILHAS**

**Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas**

São Paulo

2024

**INTEGRANTES DO PROJETO e RA’S**

Vinicius Burato - 22024055

Victor Malzone Tonioli - 23024457

Rafael Pinto da Silva - 23025211

Vinicius Massao - 23024536

Felipe Ribeiro Almeida - 23024683

Contents

[1. INTRODUÇÃO 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772436)

[2. Teste de Software 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772437)

[2.1. Apresentar 2 testes unitários. 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772438)

[2.2. Apresentar 2 testes de componentes 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772439)

[2.3. Apresentar um teste de sistema. 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772440)

[3. Qualidade de Software 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772441)

[3.1. Indicar 4 atributos de qualidade de software e informar como foi aplicado no projeto integrador (PI) 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772442)

[3.2. Apresentar um Modelo que qualidade de software 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772443)

[3.3. Apresentar um Processo (plano) de gerenciamento de qualidade de software 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772444)

[4. REFERÊNCIAS BIBLIOGRÁFICAS 3](file:///C:\Users\testelab\Downloads\Seção%20000%20Entrega%20ATIVIDADE_PI_V0.docx#_Toc164772445)

# INTRODUÇÃO

O aplicativo EstudaMILHAS permite que os usuários programem suas rotinas de estudo de forma personalizada e facilita a gestão de horas complementares e estudos semanais. O aplicativo visa promover o aprendizado contínuo em qualquer fase da vida, não se limitando apenas a universitários. O EstudaMILHAS oferece uma experiência abrangente e motivadora com recursos como a definição de metas de estudo e colaboração com instituições de ensino. Além disso, sistemas que monitoram a frequência de uso e oferecem recompensas, como "milhas de aprendizagem", incentivam os usuários a permanecerem engajados em sua jornada educacional.

# Teste de Software

# Apresentar 2 testes unitários.

1. Teste de Validação de Cálculo de Horas Complementares:

Este exame unitário avalia a conformidade do cálculo das horas complementares com as regras estabelecidas. Isso inclui fornecer um conjunto de dados representativos, como a carga horária dos cursos concluídos, e verificar se os resultados do aplicativo correspondem ao esperado. Por exemplo, após a conclusão de um curso de 20 horas, o teste deve verificar se 20 horas foram corretamente adicionadas ao total de horas complementares do usuário.

1. Teste de Integração com Serviços de Terceiros (APIs de Redes Sociais):

A integração do aplicativo com serviços de terceiros, como redes sociais, para o compartilhamento de resultados acadêmicos é o foco deste teste. Ele verifica a comunicação esperada entre o aplicativo e a API de rede social. Isso inclui verificar se o aplicativo está enviando as informações corretas para a API (por exemplo, os detalhes do curso concluído) e lidando com possíveis respostas ou erros da API. Por exemplo, o processo pode simular uma tentativa de compartilhar um título acadêmico no LinkedIn para verificar se a solicitação foi enviada corretamente e se o aplicativo tratou a resposta da API corretamente.

# Apresentar 2 testes de componentes

1. Teste de Interface do Usuário (UI): Este teste avalia se os componentes da interface do usuário estão funcionando corretamente e se apresentam a aparência esperada. Ele pode analisar várias facetas da interação do usuário, como preenchimento de formulários, navegação entre telas e cliques em botões. Por exemplo, você pode verificar se, ao clicar no botão de adicionar um novo curso, ele funciona corretamente e se a tela de detalhes do curso é exibida após a ação. Além disso, este teste pode incluir verificações visuais para garantir que os elementos da interface estejam formatados e alinhados corretamente conforme o design determinado.
2. Teste de Gerenciamento de Estado: O teste de gerenciamento de estado examina como as partes se comportam em relação ao gerenciamento de estado da aplicação. Ele verifica se os componentes estão respondendo corretamente a mudanças de estado, como atualizações de dados ou alterações na autenticação do usuário. Por exemplo, você pode testar se a lista de cursos é atualizada automaticamente após a conclusão de um novo curso e se os componentes da interface mostram corretamente o estado atual do usuário, como exibindo o número total de horas complementares acumuladas. Além disso, para garantir que os componentes lidem adequadamente com essas condições, este teste pode incluir simulações de situações de erro, como falhas na obtenção de dados do servidor.

# Apresentar um teste de sistema.

Tese: "Desenvolvimento e Avaliação de um Sistema de Gestão de Horas Complementares e Estudos Semanais para Incentivo ao Aprendizado Contínuo"

O objetivo desta tese é desenvolver e avaliar o sistema EstudaMILHAS, um aplicativo móvel projetado para ajudar os usuários a administrar suas horas complementares e estudos semanais. O objetivo do aplicativo é aumentar o aprendizado contínuo e melhorar suas habilidades cognitivas. O sistema tem funções como definir metas de estudo, se conectar a instituições educacionais para registrar horas extras e sistemas de recompensa baseados em gamificação, como reconhecimento de frequência de uso e "milhas de aprendizagem". Por meio de estudos de caso, testes de usabilidade e análises de feedback dos usuários, a pesquisa examina a usabilidade, eficácia e impacto do sistema. Os resultados da avaliação são discutidos sobre como eles ajudam e incentivam o desenvolvimento pessoal e profissional do usuário.

# Qualidade de Software

# Indicar 4 atributos de qualidade de software e informar como foi aplicado no projeto integrador (PI)

Usabilidade: Para garantir que o aplicativo seja fácil de usar e fácil de entender para os usuários, a usabilidade foi uma prioridade no projeto integrador. Além de fornecer feedback visual para ajudar os usuários durante o uso do aplicativo, foi incluído o design de uma interface de usuário amigável com navegação fácil e fácil de entender. Por exemplo, foram realizados testes de usabilidade para descobrir e corrigir problemas de confusão ou interação com o aplicativo.

Eficiência: O termo "eficiência" refere-se à capacidade do software de executar suas tarefas de forma rápida com poucos recursos, como CPU, memória e largura de banda de rede. Inclui a otimização de processos e algoritmos para reduzir o tempo de resposta do sistema e o consumo de recursos. Análises de perfil de código e testes de desempenho são ferramentas cruciais para descobrir onde a eficiência do software pode ser melhorada.

Testabilidade: A facilidade com que o software pode ser testado para garantir sua qualidade e correção é conhecida como testabilidade. Isso inclui design de código e arquitetura de software para tornar os testes automatizados fáceis de fazer e executar, bem como fornecer aos desenvolvedores ambientes e ferramentas de teste adequados. Antes de ser colocado em produção, a testabilidade é essencial para garantir que o software seja confiável e livre de defeitos.

Desempenho: Outra preocupação significativa no projeto integrador foi o desempenho do aplicativo, principalmente devido à capacidade de lidar com grandes volumes de dados, como o registro de horas adicionais de vários usuários. Testes de carga e stress foram realizados para encontrar e corrigir possíveis falhas no sistema para melhorar o desempenho. Além disso, a otimização de código e o uso eficiente de recursos de hardware garantiram uma experiência de usuário fluida, mesmo em condições de alta demanda.

# Apresentar um Modelo de qualidade de software

O Modelo de Qualidade de Produto de Software (ISO/IEC 25010) é uma norma internacional que define um conjunto de características de qualidade de software e subcaracterísticas relacionadas. Esse modelo seria adequado para o projeto EstudaMILHAS. Uma adaptação desse modelo específico para o projeto está aqui:   
  
Funcionalidade: Esta característica foca na capacidade do software de fornecer as funcionalidades que os usuários precisam para atingir seus objetivos. Para o EstudaMILHAS, isso inclui ferramentas para definir metas de estudo, registrar horas extras, receber alertas de cursos de interesse e compartilhar resultados acadêmicos nas redes sociais.

# Apresentar um Processo (plano) de gerenciamento de qualidade de software

1. Definir Objetivos: Definir os objetivos que queremos alcançar usando o software.

2. Fazer Correções: Corrigir problemas quando surgirem.

3. Testar regularmente: Nos certificar de que o software funcione como esperado.

4. Melhorar Continuamente: Procurar sempre novas formas de melhorar o software.

5. Manter a Comunicação: Manter todos atualizados sobre os avanços e mudanças.

# 4. REFERÊNCIAS BIBLIOGRÁFICAS

# Quality.Com Solutions, Seven Quality Management principles. Publicado em 14 de novembro, 2018. Acesso ao vídeo: [Seven Quality Management principles (youtube.com)](https://www.youtube.com/watch?v=TivDqV6GG9M)

Blog dos desenvolvedores Martin Fowler e Clean Code. **Links de acesso**: [martinfowler.com](https://martinfowler.com/) e [Clean Coders : Level up your code.](https://cleancoders.com/)

IEEE Software, **IET**. Publicado em 06 de agosto, 2002. **Link de acesso**: [Project infrastructure | IET Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/494935)

IEEE Software, **IEEE**. Publicado em 11 de julho, 2005. **Link de acesso**: [Improving software development management through software project telemetry | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/document/1463212)

IEEE Software, **IEEE**. Publicado em 10 de janeiro, 1984. **Link de acesso**: [Guest Editorial: Software Engineering Project Management | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/document/5010192/references#references)