



Gyga Fit

Engenharia de Software

Professor: Edeilson Milhomem

Integrantes:

- Heitor Fernandes
- João Hott
- Samuel Andrade
- Maurício Monteiro
- Hallef KayK
- Sophia Ribeiro

A Essência da APG-1 e APG-2

APG-1: Fundamentos Essenciais

- Análise de requisitos
- Planejamento inicial
- Modelagem de sistemas

APG-2: Aplicação e Evolução

- Desenvolvimento iterativo
- Testes e validação
- Entrega de valor

As APGs foram cruciais para a compreensão do ciclo de vida do software e a aplicação prática de conceitos de engenharia.

Aprendizagem Baseada em Projetos e Metodologias Ágeis

Imersão Prática

Aplicação de conceitos em cenários reais de projeto.

Feedback Contínuo

Melhoria incremental através de revisões e validações.



Flexibilidade Ágil

Adaptação a mudanças e entregas contínuas.

Colaboração Ativa

Trabalho em equipe e comunicação constante.

A experiência de projeto reforçou a importância das metodologias ágeis para o sucesso no desenvolvimento de software.

Lições e Desafios da Disciplina



Lições Aprendidas

- Gestão de escopo
- Qualidade do código
- Comunicação eficaz



Desafios Enfrentados

- Integração de equipes
- Estimativa de tempo
- Manutenção de requisitos

A disciplina nos preparou para lidar com a complexidade e os imprevistos do ambiente real de desenvolvimento.

Avaliação da Disciplina

Pontos Positivos

- Ênfase em projetos práticos
- Metodologias ágeis
- Colaboração em grupo
- Feedback construtivo

Pontos Negativos

- Necessidade de boa gestão de tempo
- Curva de aprendizado inicial
- Gestão de expectativas

Apesar dos desafios, a disciplina proporcionou um aprendizado valioso e relevante para nossa formação.

Engenharia de Software: De 0 a 10



Nota Geral do grupo: 8,5

0 Projeto Gyga Fit

GYGA FIT

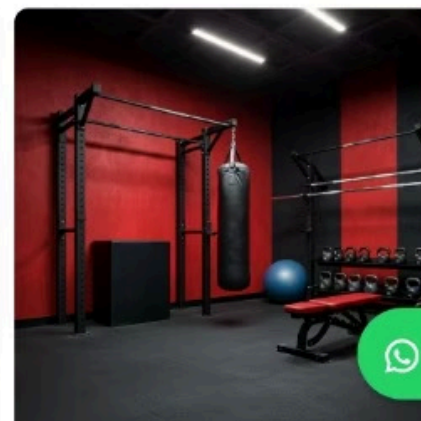
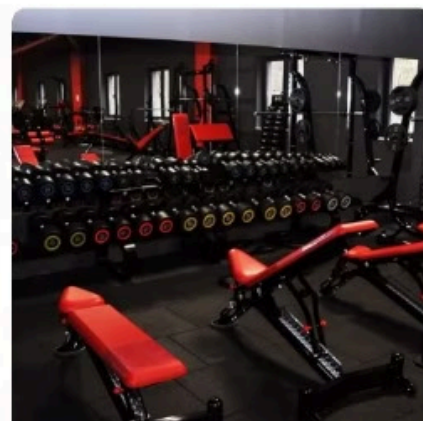
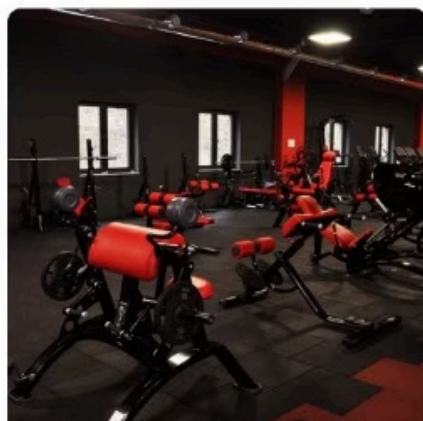
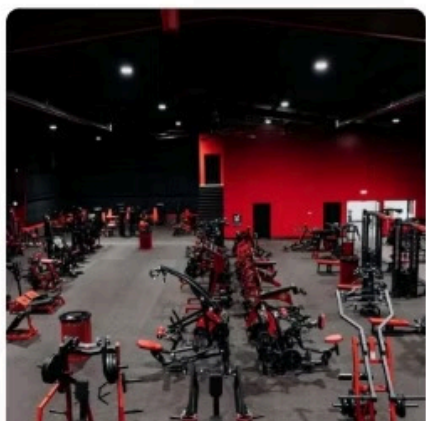
[Sobre](#) [Unidades](#) [Professores](#) [Aulas](#) [Horários](#) [Endereço](#)

[Login](#)

[Inscreva-se na Academia](#)

Sobre a Gyga Fit

A Gyga Fit é uma academia moderna com foco em bem-estar, performance e saúde. Unidades equipadas e profissionais qualificados.



Fale Conosco



Made with **GAMMA**

Sugestões de Melhoria para a Disciplina

Mais Ferramentas Práticas

Introduzir mais ferramentas de gestão de projetos e controle de versão.

Estudos de Caso Reais

Analisar projetos de software de grande escala e seus desafios.

Maior Aprofundamento de conceitos

Aprofundar mais em alguns conceitos como APIs antes de serem exigidas na implementação.

Acreditamos que estas sugestões podem enriquecer ainda mais a experiência de aprendizado.

Entendendo Problemas e Requisitos

Observação Ativa

Observação direta do contexto e dos usuários para compreender melhor o problema a ser resolvido.

1

2

Entrevistas com Stakeholders

Conversas com as partes interessadas para coletar suas perspectivas e requisitos.

Análise Documental

Revisão de documentos, especificações e materiais existentes para entender o problema em profundidade.

3

4

Prototipagem Colaborativa

Desenvolvimento de protótipos com os usuários para validar e refinar os requisitos.

Priorizando Requisitos do Produto

Matriz de Risco/Valor

Utilizaremos uma matriz de risco e valor para priorizar os requisitos mais importantes e com menor risco de implementação.

Feedback do Usuário

Coletaremos feedback contínuo dos usuários-chave para validar os requisitos e ajustar a priorização conforme necessário.

- 1. Quais estratégias utilizaríamos para entender o problema e todos os seus requisitos?**
- 2. Como definiríamos os requisitos prioritários do produto que precisa ser desenvolvido?**
- 3. Como validaríamos os requisitos?**
- 4. Como definiríamos a definição e validação das tecnologias que seriam adotadas?**

- 5. Qual a importância do planejamento das iterações?**
- 6. Qual a importância da definição de valor em uma iteração para o sucesso do produto?**
- 7. Qual a importância de seguir padrões em projetos, por exemplo, MVC?**
- 8. Qual a importância de seguir a Arquitetura baseada em Serviços por meio de API REST?**

- 9. Vocês conseguem visualizar a importância de Testes Unitários em projetos de Software?**
- 10. Qual a importância do Controle de Versão do código? Como este controle de versão é realizado?**
- 11. Como entendemos o fluxo de trabalho organizado pelo GitFlow?**
- 12. Como entendemos a release do produto?**

Obrigado