

Cronograma do Projeto

- **14/05 - Desvendando o Desenvolvimento Web: HTML, CSS e JavaScript**
 - ✓ **Palestrante:** Rodrigo Costa, Front-End na Motoca Systems, Líder de Front-end na SCS.
 - ✓ **Atividades:** Exploração das melhores práticas em HTML, CSS e JavaScript.
- **21/05 - Apresentação das Padronizações no Projeto e no Jira**
 - ✓ **Palestrante:** Gerente de Projetos.
 - ✓ **Atividades:** Demonstração das padronizações adotadas no projeto e como elas foram implementadas no Jira.
- **21/05 - Apresentações Individuais da Equipe**
 - ✓ **Atividades:** Cada membro da equipe deverá se apresentar, explicar sua função no projeto e discutir as tarefas que realizou.
 - ✓ **Duração:** 10 minutos para cada apresentação.

Observações aos Alunos:

1. **Padronização das Tarefas:**
 - ✓ **1.1 Conteúdo Dentro das Tarefas:** Cada tarefa deve incluir descrições claras do que precisa ser feito, especificando tecnologias e recursos necessários.
2. **Sprints com Nomes Sugestivos:**
 - ✓ **2.1 Tarefas Dentro das Sprints:** Organize as tarefas em sprints com nomes que reflitam claramente os objetivos e o trabalho a ser realizado durante aquele período.
3. **Epics Padronizados:**
 - ✓ **3.1 Tarefas Vinculadas aos Epics:** Cada tarefa deve ser associada a um epic que agrupa várias tarefas relacionadas a um objetivo maior do projeto.
4. **Documentação Pronta:**
 - ✓ **4.1 Documentação sobre o Projeto:** Elabore uma documentação completa do projeto, incluindo requisitos, design, arquitetura, e detalhes de implementação.
 - ✓ **4.2 Mencionando Tempo de Desenvolvimento:** Registre e documente o tempo de desenvolvimento previsto e real para cada tarefa, facilitando a análise de produtividade e planejamento futuro.

Ferramentas Recomendadas:

- **Jira ou Trello:** Para gerenciamento de tarefas, sprints, e epics.
- **Google Docs ou Confluence:** Para criação e compartilhamento de documentação colaborativa.

CONTEUDO DO DOCUMENTO DO PROJETO

1. Definição de Requisitos:

- **Requisitos Funcionais:** Descrevem as funcionalidades que o software deve oferecer.
- **Requisitos Não Funcionais:** Incluem aspectos como desempenho, segurança, portabilidade e usabilidade.

2. Planejamento:

- **Cronograma:** Definição das fases do projeto e suas respectivas durações.
- **Orçamento:** Estimativa dos custos associados ao desenvolvimento.
- **Recursos:** Alocar pessoal, ferramentas e outros recursos necessários.

3. Análise e Projeto de Sistema:

- **Modelagem:** Uso de diagramas e modelos para representar o sistema, como UML.
- **Arquitetura de Software:** Definir a estrutura do sistema, incluindo módulos e interfaces.

4. Implementação:

- **Codificação:** Escrita do código fonte com base no projeto definido.
- **Ferramentas de Desenvolvimento:** Selecionar linguagens de programação, frameworks e ambientes de desenvolvimento.

5. Testes:

- **Testes Unitários:** Verificar cada componente ou módulo individualmente.
- **Testes de Integração:** Testar a combinação de módulos para verificar a interação entre eles.
- **Testes de Sistema:** Avaliar o sistema completo para garantir que atende aos requisitos especificados.
- **Testes de Aceitação:** Realizados com a participação do cliente para garantir que o sistema faz o que eles precisam.

6. Implantação:

- **Estratégias de Lançamento:** Definir como o software será entregue aos usuários finais.
- **Treinamento e Documentação:** Preparar manuais do usuário e treinar os usuários no uso do software.

7. Manutenção:

- **Correções e Atualizações:** Continuar a desenvolver o software após a implantação para corrigir falhas e responder a novos requisitos ou problemas.

8. Gerenciamento de Projeto:

- **Metodologias:** Adotar uma metodologia de gerenciamento de projetos, como Scrum, Kanban ou Waterfall.

- **Ferramentas de Gerenciamento:** Utilizar ferramentas para controle de versões, acompanhamento de tarefas e colaboração.

9. Documentação:

- **Documentação Técnica:** Incluir especificações de design, comentários no código e descrições de API.
- **Documentação do Usuário:** Manuais e guias para ajudar os usuários a entender e utilizar o software.