  
**Flutter – Gestão de gerenciamento de projetos de TI**

**Professor: Diego**

**Aluno: David Luis Kim**

**Atividade EAD: Criação de uma Estrutura Analítica de Projeto (EAP)**

**Data de entrega:** Aberto: 07/08/2025 Vencimento: 14/08/2025

**Integrantes do Grupo:**

1. André Felipe I Leite - Developer
2. Daniel Vinhais Greenwood - Developer
3. David Luis Kim – Scrum Master
4. Luiz Threis – Product Owner

**Descrição da Atividade:**

No universo da Tecnologia da Informação, o sucesso de um projeto depende não apenas da capacidade de codificar ou configurar sistemas, mas principalmente da **forma como esse projeto é estruturado, planejado e monitorado ao longo do tempo**. Um dos maiores desafios enfrentados por profissionais da área é justamente o de transformar uma ideia inicial em um plano claro e organizado, que envolva **tarefas bem definidas, entregas visíveis e responsáveis identificadas**. É nesse ponto que a **Estrutura Analítica de Projeto (EAP)** se torna uma ferramenta essencial.

Segundo o *Project Management Institute (2021)*, a EAP — também conhecida como Work Breakdown Structure (WBS) — é a decomposição hierárquica orientada à entrega de todos os componentes de um projeto. Ela permite visualizar de forma estruturada todas as partes do trabalho que precisam ser executadas, facilitando o controle de escopo, tempo, custos e recursos. O próprio PMBOK destaca que uma EAP bem construída “melhora a comunicação entre os membros da equipe, define limites claros de responsabilidade e facilita a estimativa de cronograma e custos” (PMI, 2021, p. 162).

Por exemplo, imagine um projeto para desenvolver um **aplicativo de gerenciamento de tarefas**. Sem uma estrutura adequada, é comum que a equipe enfrente problemas como retrabalho, funcionalidades mal definidas ou prazos estourados. No entanto, ao aplicar a técnica da EAP, é possível quebrar o projeto em entregas principais, como “Interface do Usuário”, “Banco de Dados”, “Funcionalidades de Lembrete” e “Testes”. Cada uma dessas entregas pode ser subdividida em **pacotes de trabalho menores**, como “Criar tela de login”, “Desenhar wireframe”, “Implementar autenticação com Firebase” ou “Criar casos de teste automatizados”. Isso favorece o acompanhamento do progresso e o alinhamento entre todos os envolvidos.

Autores como Pressman (2016) e Sommerville (2019) concordam que **um dos principais fatores de falha em projetos de software é a falta de organização clara e de escopo bem definido**. A EAP, nesse sentido, é uma solução eficaz e amplamente reconhecida no setor. Além disso, o exercício de elaborar uma EAP em equipe contribui para o desenvolvimento de habilidades essenciais no mercado atual, como **comunicação, colaboração, raciocínio lógico e visão sistêmica**.

O domínio dessa ferramenta também possibilita a **transição entre abordagens tradicionais e ágeis**. Enquanto o PMBOK a utiliza como instrumento clássico para controle de escopo, metodologias como o Scrum podem se beneficiar de uma EAP adaptada, usada como base para organizar sprints, tarefas do backlog e fluxos de entrega contínua (*Bezerra, 2022*).

Portanto, nesta atividade, você e sua equipe deverão **elaborar uma EAP completa para um projeto de software**, aplicando, conceitos aprendidos e aproximando-se da realidade profissional de empresas que desenvolvem sistemas sob demanda.

**Situação-Problema:**

Você faz parte de uma equipe de analistas e desenvolvedores contratada por uma empresa que precisa implementar um **novo sistema digital personalizado**. O cliente já possui a ideia geral, mas não tem clareza sobre como dividir esse projeto em etapas práticas e controláveis. A direção da empresa solicita que vocês elaborem uma **estrutura visual e descritiva do projeto** — ou seja, uma EAP — que mostre claramente:

* Os **blocos principais do sistema**;
* As **atividades específicas dentro de cada bloco**;
* Os **responsáveis por cada etapa**;
* Os **critérios que indicam que uma tarefa foi concluída com sucesso**.

A EAP será o documento base para **definir prazos, atribuir responsabilidades e controlar a execução do projeto**, e deve estar organizada de forma hierárquica e compreensível

**Comando da Atividade:**

Com base na situação descrita, sua equipe deverá:

**1 - Escolher e definir um projeto de software**  
**a)** Escolha um dos projetos abaixo ou proponha outro, com aprovação do professor.

|  |
| --- |
| Aplicativo de gerenciamento de tarefas – **Time Room** (Sistema de reservas de salas de reunião). |

**b)** Escreva uma descrição clara do objetivo do sistema e liste suas funcionalidades principais.

|  |
| --- |
| O **Time Room** é um sistema digital que permitirá aos funcionários realizarem a reserva de salas de reunião de forma rápida, prática e organizada. O sistema garantirá que não haja conflitos de horário, permitindo ao usuário selecionar a sala desejada, informar a data, o horário de início e fim da reunião, bem como o número de participantes. Além disso, a solução enviará confirmações por e-mail, assegurando clareza e confiabilidade no processo de agendamento.  **Estrutura geral do Wireframe – Time Room**   * **Tela de Login**   + Campos: E-mail, Senha   + Botão: Entrar   + Link: “Esqueci minha senha”   + Link: “Criar conta” * **Tela de Cadastro**   + Nome completo, E-mail, Senha, confirmar senha   + Botão: Cadastrar * **Tela Inicial (Dashboard)**   + Saudações personalizadas: “Olá, [Nome]!”   + Botão: Nova Reserva   + Lista das reservas atuais (sala, data, horário, status)   + Botão flutuante: Ver Salas Disponíveis * **Tela de Nova Reserva**   + Selecionar sala (dropdown)   + Data (calendar picker)   + Horário de início/fim (time picker)   + Número de participantes (campo numérico)   + Botão: Reservar   + Mensagem de erro em caso de conflito de horário * **Tela de Confirmação**   + Exibe os dados da reserva   + Botão: Confirmar e enviar por e-mail   + Mensagem: “Reserva feita com sucesso! Um e-mail foi enviado.” * **Tela de Salas Disponíveis**   + Lista de salas com nome, capacidade e status de disponibilidade   + Botão: Reservar agora (leva à tela de nova reserva já preenchida com a sala) * **Tela de Perfil (opcional)**   + Dados do usuário   + Botão: Editar dados   + Botão: Sair |

c) Liste os requisitos essenciais do sistema (funcionais e, se possível, não funcionais).

|  |
| --- |
| **Requisitos Funcionais**   * O funcionário deve poder se autenticar (login/cadastro). * O sistema deve permitir criar, editar e cancelar reservas de salas. * O sistema deve garantir que não haja conflitos de horário. * A reserva deve incluir sala, data, horário de início e fim, e número de participantes. * O funcionário deve receber uma confirmação da reserva por e-mail. * O sistema deve listar salas disponíveis, incluindo capacidade e status. * O usuário deve ter acesso a suas reservas atuais.   **Requisitos Não Funcionais**   * A interface deve ser responsiva e intuitiva (acesso via web e mobile). * O tempo de resposta do sistema deve ser inferior a 2 segundos para operações críticas. * O sistema deve suportar autenticação segura com criptografia de senhas. * Deve haver registro de logs de reservas para auditoria. * O sistema deve estar disponível pelo menos 99% do tempo (alta disponibilidade).   **Critérios de Aceitação (exemplos)**   * O funcionário deve poder selecionar apenas salas disponíveis no sistema. * O sistema deve impedir reservas conflitantes em datas/horários já ocupados. * Após confirmação, a reserva deve exibir os dados completos. * O funcionário deve receber automaticamente uma notificação por e-mail com os detalhes. |

**2- Criar a Estrutura Analítica de Projeto (EAP)**  
a) Organize a EAP até **pelo menos o Nível 3**, seguindo a hierarquia:

|  |
| --- |
| **Nível 1: Projeto**  Sistema Timeroom – Reserva de Salas  **Nível 2: Entregas Principais**   1. Autenticação e Usuários 2. Gestão de Salas 3. Reservas e Agendamento 4. Relatórios e Monitoramento 5. Interface do Usuário (Front-end) 6. Integração e Banco de Dados 7. Infraestrutura e Deploy 8. Gestão do Projeto   **Nível 3: Pacotes de Trabalho**  1. Autenticação e Usuários   * 1.1 Criar tela de login * 1.2 Implementar cadastro de usuários * 1.3 Configurar níveis de permissão (admin, usuário comum) * 1.4 Recuperação de senha   2. Gestão de Salas   * 2.1 Cadastro de salas (nome, capacidade, recursos disponíveis) * 2.2 Edição e exclusão de salas * 2.3 Visualização de disponibilidade   3. Reservas e Agendamento   * 3.1 Criar tela de reserva de sala * 3.2 Implementar sistema de checagem de disponibilidade * 3.3 Cancelamento e edição de reservas * 3.4 Notificações de confirmação/cancelamento   4. Relatórios e Monitoramento   * 4.1 Relatório de uso por sala * 4.2 Relatório de reservas por período * 4.3 Dashboard de utilização em tempo real   5. Interface do Usuário (Front-end)   * 5.1 Design responsivo * 5.2 Navegação intuitiva * 5.3 Testes de usabilidade   6. Integração e Banco de Dados   * 6.1 Modelagem do banco de dados * 6.2 Integração com API * 6.3 Otimização de consultas   7. Infraestrutura e Deploy   * 7.1 Configuração do ambiente de desenvolvimento * 7.2 Deploy em servidor/Cloud * 7.3 Backup e segurança dos dados   8. Gestão do Projeto   * 8.1 Planejamento e cronograma * 8.2 Acompanhamento de sprints (Scrum/Kanban) * 8.3 Documentação do sistema |

b) Se necessário, crie um **Nível 4** para detalhar tarefas específicas.

|  |
| --- |
| 1. Sistema de Reservas de Salas (Timeroom)   1.2 Gestão de Usuários     1.2.1 Criar tela de login       1.2.1.1 Implementar formulário de login       1.2.1.2 Validar credenciais no banco de dados       1.2.1.3 Implementar recuperação de senha  1.3 Reserva de Salas     1.3.2 Implementar reserva de sala       1.3.2.1 Criar tela de agendamento       1.3.2.2 Validar disponibilidade em tempo real       1.3.2.3 Gravar reserva no banco de dados  1.4 Relatórios e Monitoramento     1.4.2 Criar relatórios de utilização       1.4.2.1 Relatório diário de reservas       1.4.2.2 Relatório semanal de ocupação       1.4.2.3 Relatório mensal em PDF |

c) Apresente a EAP em **formato de árvore (gráfico)** ou **tabela estruturada**, usando ferramentas como Excel, PowerPoint, Word, Miro, Trello ou outra combinada com o professor.

|  |
| --- |
| 1. Sistema de Reservas de Salas (Timeroom)  │  ├── 1.1 Configuração Inicial  │ ├── 1.1.1 Levantamento de requisitos  │ ├── 1.1.2 Definição da arquitetura  │ └── 1.1.3 Configuração do banco de dados  │  ├── 1.2 Gestão de Usuários  │ ├── 1.2.1 Criar tela de login  │ │ ├── 1.2.1.1 Implementar formulário  │ │ ├── 1.2.1.2 Validar credenciais  │ │ └── 1.2.1.3 Recuperação de senha  │ ├── 1.2.2 Cadastro de novos usuários  │ └── 1.2.3 Gestão de permissões  │  ├── 1.3 Reserva de Salas  │ ├── 1.3.1 Criar interface de busca  │ ├── 1.3.2 Implementar reserva  │ │ ├── 1.3.2.1 Tela de agendamento  │ │ ├── 1.3.2.2 Validar disponibilidade  │ │ └── 1.3.2.3 Gravar no banco  │ └── 1.3.3 Cancelamento/edição de reserva  │  ├── 1.4 Relatórios e Monitoramento  │ ├── 1.4.1 Painel de visualização  │ └── 1.4.2 Relatórios  │ ├── 1.4.2.1 Diário  │ ├── 1.4.2.2 Semanal  │ └── 1.4.2.3 Mensal em PDF  │  └── 1.5 Infraestrutura e Segurança  ├── 1.5.1 Backup de dados  ├── 1.5.2 Certificado SSL  └── 1.5.3 Controle de acessos |

**3- Descrever os elementos de cada tarefa da EAP**

|  |
| --- |
| Desenvolvimento do Sistema  1 – Backend  1.1 – Estrutura do banco de dados   * + Descrição: Criação do banco de dados no Supabase com tabelas para usuários, reservas e salas.   + Responsável: Dev Backend   + Critérios de aceitação: O banco contém todas as tabelas normalizadas e integradas ao sistema; é possível cadastrar, atualizar e consultar dados.   1.2 – API de autenticação   * + Descrição: Implementar endpoints para login, cadastro e recuperação de senha usando Supabase Auth.   + Responsável: Dev Backend   + Critérios de aceitação: O usuário consegue se cadastrar, logar e recuperar senha; tentativas inválidas retornam erro.   1.3 – API de reservas   * + Descrição: Criar endpoints REST para criar, editar, cancelar e listar reservas de salas.   + Responsável: Dev Backend   + Critérios de aceitação: O sistema garante que reservas não se sobrepõem; confirmações são gravadas no banco.   Frontend  2.1 – Tela de login/cadastro   * + Descrição: Interface em Flutter para login, cadastro e recuperação de senha.   + Responsável: Dev Frontend   + Critérios de aceitação: O usuário acessa o sistema após autenticação válida; erros exibem mensagens amigáveis.   2.2 – Dashboard de salas   * + Descrição: Exibir lista de salas disponíveis com calendário e horários.   + Responsável: Dev Frontend   + Critérios de aceitação: Usuário visualiza salas, horários ocupados e horários livres.   2.3 – Tela de reserva   * + Descrição: Tela para selecionar sala, data e horário, confirmando reserva.   + Responsável: Dev Frontend   + Critérios de aceitação: Usuário consegue reservar uma sala e ver a confirmação no dashboard. |

**4- Definir os critérios de aceitação**

|  |
| --- |
| Critérios de Aceitação – Timeroom (Sistema de Reserva de Salas)  Design da Interface do Usuário   * A tela inicial apresenta opções claras de login, cadastro e visualização de reservas. * Layout responsivo funcionando corretamente em desktop e dispositivos móveis. * Ícones, botões e textos seguem o padrão de design definido.   Implementação da Tela de Login   * O sistema autentica corretamente com credenciais válidas. * O sistema rejeita credenciais inválidas e exibe mensagem clara de erro. * Após login bem-sucedido, o usuário é redirecionado ao painel principal.   Implementação do Cadastro de Usuários   * O formulário salva dados corretamente no banco de dados. * Campos obrigatórios não permitem envio em branco. * Senhas seguem regras de segurança (mínimo de caracteres, caracteres especiais etc.). * E-mail válido é verificado e duplicidade não é permitida.   Implementação da Reserva de Salas   * O usuário consegue selecionar data, horário e sala disponíveis. * O sistema bloqueia reservas em horários já ocupados. * Reserva é confirmada e salva no banco de dados. * Usuário recebe feedback de sucesso da operação.   Cancelamento de Reservas   * O usuário consegue visualizar suas reservas e escolher qual cancelar. * Após cancelamento, a sala volta a ficar disponível para outros usuários. * O sistema confirma o cancelamento com mensagem ao usuário.   Visualização de Agenda   * O calendário exibe corretamente todas as reservas ativas. * Atualizações em tempo real após criação ou cancelamento de reservas. * Interface permite navegação entre dias/meses.   Integração com Banco de Dados   * Todas as operações de login, cadastro, reserva e cancelamento gravam e leem dados corretamente do banco. * Em caso de falha de conexão, o sistema exibe mensagem amigável de erro. * Dados são persistidos corretamente e não se perdem após reinício.   Testes de Funcionalidade   * Cada módulo (login, cadastro, reserva, cancelamento, visualização) é testado individualmente. * Os testes confirmam que as regras de negócio foram atendidas. * Nenhum erro crítico é encontrado após a bateria de testes. |

**5- Produto Final Esperado**

* EAP completa do projeto de software (em formato gráfico ou tabela);
* Lista de tarefas com código, nome, descrição e responsável;
* Documento contendo os critérios de aceitação organizados por tarefa.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EAP COMPLETA**   |  |  |  |  | | --- | --- | --- | --- | | **Código** | **Entrega Principal / Tarefa** | **Descrição** | **Responsável** | | 1 | Sistema Timeroom | Projeto completo de reservas de salas | Equipe inteira | | 1.1 | Configuração Inicial | Levantamento de requisitos, arquitetura, DB | Equipe Back | | 1.2 | Gestão de Usuários | Criação e manutenção de contas de usuários | Equipe Front/Back | | 1.2.1 | Tela de Login | Interface para login de usuários | David / André | | 1.2.2 | Cadastro de Usuários | Formulário e backend para cadastro | Felipe / Luiz | | 1.2.3 | Recuperação de Senha | Funcionalidade de reset de senha | Daniel / David | | 1.3 | Reserva de Salas | Gerenciamento completo de reservas | Equipe Front/Back | | 1.3.1 | Tela de Nova Reserva | Seleção de sala, data, horário, nº de pessoas | David / André | | 1.3.2 | Validação de Conflito | Bloquear reservas conflitantes | Felipe / Daniel | | 1.3.3 | Confirmação de Reserva | Envio de e-mail e feedback ao usuário | Luiz / David | | 1.4 | Relatórios e Monitoramento | Painel de reservas e relatórios de uso | Daniel / Felipe | | 1.4.1 | Relatórios Diários/Semanais/Mensais | Gerar relatórios em PDF ou visualização | Luiz / André | | 1.5 | Infraestrutura e Deploy | Configuração de servidor, backup, SSL | Equipe Back | | 1.6 | Testes e Validação | Testes unitários e integração | QA / Todos |   **LISTA DE TAREFAS**   |  |  |  |  | | --- | --- | --- | --- | | **Código** | **Nome da Tarefa** | **Descrição detalhada** | **Responsável** | | 1.2.1 | Tela de Login | Desenvolver interface de login com validação de campos | David / André | | 1.2.2 | Cadastro de Usuários | Criar formulário e integrar ao backend | Felipe / Luiz | | 1.2.3 | Recuperação de Senha | Implementar envio de link de recuperação por e-mail | Daniel / David | | 1.3.1 | Tela de Nova Reserva | Seleção de sala, data, horário e número de participantes | David / André | | 1.3.2 | Validação de Conflito | Bloquear horários já reservados | Felipe / Daniel | | 1.3.3 | Confirmação de Reserva | Enviar e-mail de confirmação e mostrar mensagem de sucesso | Luiz / David | | 1.4.1 | Relatórios Diários/Semanais/Mensais | Gerar relatórios de uso e reservas | Luiz / André | | 1.5 | Infraestrutura e Deploy | Configurar servidor, backup e SSL | Equipe Back | | 1.6 | Testes e Validação | Testes unitários, integração e funcionalidade | QA / Todos |   **CRITÉRIOS DE ACEITAÇÃO**   |  |  |  | | --- | --- | --- | | **Código** | **Tarefa** | **Critério de Aceitação** | | 1.2.1 | Tela de Login | Login válido permite acesso; login inválido exibe mensagem; redireciona corretamente | | 1.2.2 | Cadastro de Usuários | Campos obrigatórios validados; duplicidade de e-mail bloqueada; senha segura | | 1.2.3 | Recuperação de Senha | Usuário recebe e-mail com link válido; senha pode ser redefinida | | 1.3.1 | Tela de Nova Reserva | Usuário seleciona sala, data e horário; reserva é salva no sistema | | 1.3.2 | Validação de Conflito | Sistema bloqueia horários conflitantes; alerta de erro exibido | | 1.3.3 | Confirmação de Reserva | Mensagem de sucesso exibida; e-mail de confirmação enviado | | 1.4.1 | Relatórios Diários/Semanais/Mensais | Relatórios gerados corretamente; dados conferem com reservas existentes | | 1.5 | Infraestrutura e Deploy | Sistema disponível no servidor; SSL ativo; backups funcionando | | 1.6 | Testes e Validação | Todos os módulos testados; nenhum erro crítico; fluxos principais funcionam | |