

## Proposta de Trabalho - Projeto Aplicado Multiplataforma Etapa 1 (N705)

### Planejamento e Arquitetura de Sistema Multiplataforma

#### Conhecimentos necessários para o desenvolvimento da atividade

Para o desenvolvimento desta atividade, os alunos deverão possuir ou desenvolver os seguintes conhecimentos:

- Fundamentos de arquitetura de software
- Design e modelagem de sistemas multiplataforma
- Princípios de UX/UI para diferentes plataformas
- Modelagem de banco de dados
- Padrões arquitetrais para aplicações web e móveis
- Documentação técnica e de requisitos
- Prototipação de interfaces
- Conhecimentos na linguagem de marcação markdown

#### Objetivo da proposta de atividade

Esta atividade tem como objetivo desenvolver habilidades de planejamento e arquitetura de sistemas multiplataforma. Os alunos deverão aplicar conceitos de engenharia de software para projetar uma solução completa, definindo sua arquitetura, modelando banco de dados, criando protótipos de interface e elaborando documentação técnica detalhada. **Esta etapa foca exclusivamente no planejamento e documentação, sem implementação de código, que será realizada na Etapa 2 (disciplina N708).**

Os alunos deverão identificar e escolher uma demanda real da sociedade (empresa, ONG, organização da sociedade civil, órgão público, etc.) que possa ser atendida por meio de uma solução de software, buscando gerar impacto social positivo através da tecnologia.

#### Descrição da atividade

Os alunos deverão projetar um sistema multiplataforma completo, elaborando toda a documentação técnica necessária para seu desenvolvimento futuro na

Etapa 2 (N708). O projeto deve resolver um problema real e demonstrar a aplicação prática dos conceitos de arquitetura e planejamento de software estudados nas disciplinas do curso até o momento.

### **Requisitos técnicos obrigatórios:**

1. Elaborar documento de requisitos funcionais e não-funcionais
2. Definir a arquitetura completa do sistema (frontend, backend, APIs, banco de dados)
3. Modelar o banco de dados com diagrama ER ou equivalente
4. Criar protótipos de interface para web e dispositivos móveis
5. Documentar as APIs e serviços planejados
6. Definir tecnologias e ferramentas a serem utilizadas na implementação futura
7. Planejar estrutura de testes e validação
8. Elaborar cronograma de desenvolvimento para a Etapa 2 (N708)

### **Exemplos de Sistemas com Impacto Social em Fortaleza (apenas para despertar ideias)**

A seguir, apresentamos exemplos de sistemas que podem atender a demandas sociais específicas de Fortaleza e seus bairros:

1. **Sistema de Gestão para Abrigos de Animais:** Plataforma para gerenciamento de adoções, doações, voluntariado e controle de estoque para abrigos de animais abandonados em Fortaleza, como os localizados no bairro Passaré e Jangurussu.
2. **Plataforma de Educação Complementar para Escolas Públicas:** Sistema que conecte voluntários, professores e estudantes para oferta de aulas de reforço e atividades educativas complementares em escolas públicas de bairros como Bom Jardim, Granja Portugal e Conjunto Palmeiras.
3. **Sistema de Gestão para Bancos Comunitários:** Plataforma para gerenciamento de microcrédito, moedas sociais e iniciativas de economia solidária para bancos comunitários como o Banco Palmas no Conjunto Palmeiras.

4. **Aplicação para Mapeamento e Monitoramento de Pontos de Descarte Irregular de Lixo:** Sistema que permita cidadãos reportarem pontos críticos de acúmulo de lixo em bairros como Pirambu, Barra do Ceará e Vicente Pinzón, auxiliando na gestão ambiental urbana.
5. **Plataforma de Apoio a Pequenos Produtores Locais:** Sistema para conectar pequenos agricultores da região metropolitana com consumidores de Fortaleza, facilitando feiras virtuais e entregas diretas, beneficiando especialmente as áreas rurais de Caucaia, Maracanaú e Eusébio.
6. **Sistema de Gestão para Associações de Reciclagem:** Plataforma para gerenciar coleta, separação e comercialização de material reciclável, beneficiando cooperativas como as do Jangurussu e do Dionísio Torres.
7. **Aplicação para Compartilhamento de Caronas Solidárias:** Sistema que conecte pessoas que fazem trajetos similares diariamente para universidades e polos de trabalho, reduzindo emissões e custos de transporte, beneficiando especialmente moradores de bairros mais distantes como Messejana e Conjunto Ceará.
8. **Plataforma para Gestão de Postos de Saúde Comunitários:** Sistema para agendamentos, prontuários simplificados e acompanhamento de campanhas de saúde em postos dos bairros como Edson Queiroz, Barroso e Passaré.
9. **Sistema de Gerenciamento para Bibliotecas Comunitárias:** Plataforma para catalogação, empréstimos e eventos culturais em bibliotecas comunitárias de bairros como Mucuripe, Serviluz e Cais do Porto.
10. **Plataforma para Mapeamento de Iniciativas Culturais Periféricas:** Sistema para divulgação e gestão de eventos culturais nas periferias, dando visibilidade a artistas locais de bairros como Grande Bom Jardim, Jangurussu e Mondubim.

### **Objetivo de Desenvolvimento Sustentável (ODS 11)**

Para este projeto, os alunos deverão trabalhar com o **ODS 11: Cidades e Comunidades Sustentáveis**.

O ODS 11 visa tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis. Esta escolha se alinha com projetos de arquitetura de software, uma vez que o planejamento de tecnologias digitais pode contribuir para a criação de cidades mais inteligentes e sustentáveis.

A avaliação considerará a relevância e o potencial impacto da solução projetada em relação às metas do ODS 11, bem como a clareza na explicação de como a arquitetura proposta pode contribuir para cidades mais sustentáveis.

Para mais informações sobre o ODS 11, consulte: <https://brasil.un.org/pt-br/sdgs/11>

### **Passo a passo para o desenvolvimento da atividade**

#### **1. Formação das equipes**

- Formar grupos de 4 a 6 integrantes (obrigatório entrar no grupo via AVA)
- Definir papéis e responsabilidades dentro da equipe (lembre-se de documentar no relatório final)

#### **2. Análise e Planejamento**

- Identificação do problema a ser resolvido
- Análise do público-alvo e stakeholders
- Levantamento de requisitos funcionais e não-funcionais

#### **3. Design da Arquitetura**

- Definição da arquitetura do sistema
- Escolha de padrões arquiteturais
- Especificação de tecnologias e frameworks

#### **4. Modelagem**

- Criação de diagramas de fluxo e componentes
- Modelagem do banco de dados
- Definição das APIs e interfaces de comunicação

#### **5. Prototipação**

- Elaboração de wireframes
- Criação de protótipos de interface
- Validação de fluxos de usuário

## 6. Documentação e Finalização

- Elaboração do documento de arquitetura
- Compilação dos artefatos técnicos
- Preparação do relatório final

### Orientações para o envio da atividade

1. Todos os integrantes do grupo devem estar devidamente cadastrados na equipe via AVA.
2. Todos os representantes do grupo devem submeter no AVA, no local apropriado para envio da atividade:
  - Documento contendo nome completo de todos os integrantes e suas respectivas matrículas
  - No documento deve conter o link para o repositório GitHub contendo a documentação do projeto
  - No documento deve constar um breve resumo do projeto
3. O repositório GitHub deve conter:
  - README.md detalhado na raiz do projeto, contendo:
    - Objetivo do projeto
    - Descrição funcional da solução planejada
    - Visão geral da arquitetura com diagrama
    - Lista de tecnologias propostas
    - Cronograma de desenvolvimento para a Etapa 2 (N708)
  - Pasta docs/ com toda a documentação técnica em arquivos markdown
  - Protótipos de interface das telas principais
4. A documentação deve ser clara e completa, permitindo que o desenvolvimento na Etapa 2 (N708) possa ser realizado baseando-se nos artefatos fornecidos.

## Estrutura do repositório (obrigatória, até este ponto)

```
/  
|   README.md          # Documentação principal do projeto  
|   docs/  
|       requirements/  
|           requirements.md      # Requisitos funcionais e não-funcionais  
|       architecture/  
|           architecture.md    # Documentação da arquitetura do sistema  
|       database/  
|           database_model.md # Modelo de dados com diagrama ER  
|       api/  
|           api_specification.md # Especificação das APIs planejadas  
prototypes/  
|   web/                # Protótipos de interface  
|       [arquivos de imagem]  
|   mobile/              # Protótipos de interface mobile  
|       [arquivos de imagem]
```

## Conteúdo obrigatório dos arquivos de documentação

### 1. README.md

- Título e descrição do projeto
- Problema abordado e justificativa
- Objetivos do sistema
- Escopo do projeto
- Visão geral da arquitetura (com diagrama)
- Tecnologias propostas
- Cronograma para Etapa 2 (N708)
- Integrantes da equipe e seus papéis

### 2. docs/requirements/requirements.md

- Requisitos funcionais (RF)
- Requisitos não-funcionais (RNF)
- Regras de negócio
- Histórias de usuário ou casos de uso
- Perfis de usuários

### 3. docs/architecture/architecture.md

- Descrição da arquitetura
- Componentes do sistema
- Padrões arquiteturais utilizados
- Diagrama de arquitetura
- Decisões técnicas e justificativas

#### 4. **docs/database/database\_model.md**

- Modelo de dados
- Descrição das entidades e relacionamentos
- Diagrama ER ou similar
- Dicionário de dados

#### 5. **docs/api/api\_specification.md**

- Endpoints previstos
- Parâmetros de requisição
- Formatos de resposta
- Autenticação e autorização
- Exemplos de chamadas e respostas

### **Critérios de avaliação (pontuação máxima de 5 pontos)**

#### 1. **Documentação de Requisitos** (1,5 pontos)

- Clareza e detalhamento dos requisitos funcionais e não-funcionais (0,6)
- Adequação da solução ao problema identificado (0,5)
- Escopo bem definido e histórias de usuário claras (0,4)

#### 2. **Arquitetura e Modelagem** (2,0 pontos)

- Qualidade e consistência da arquitetura proposta (0,7)
- Adequação das tecnologias escolhidas (0,5)
- Qualidade do modelo de dados e especificação de APIs (0,8)

#### 3. **Prototipação e Planejamento** (1,5 pontos)

- Qualidade e usabilidade dos protótipos de interface (0,7)
- Organização da documentação técnica (0,4)
- Viabilidade do cronograma proposto para a Etapa 2 (N708) (0,4)

### **Checklist para submissão**

Para garantir que seu projeto atenda aos requisitos para avaliação, verifique se:

- O documento de requisitos (**requirements.md**) inclui requisitos funcionais e não-funcionais claramente definidos
- A arquitetura do sistema está claramente definida e diagramada (**architecture.md**)

- O modelo de dados está completo com relacionamentos (database\_model.md)
- As APIs estão especificadas com endpoints, parâmetros e respostas (api\_specification.md)
- Os protótipos de interface cobrem as principais funcionalidades do sistema
- A documentação explica como o projeto se relaciona ao ODS 11
- O cronograma para a Etapa 2 (N708) está detalhado no README.md
- O repositório segue exatamente a estrutura de pastas e arquivos definida

### Total de pontos

A atividade tem valor total de 5,0 pontos, que correspondem a 50% da nota do trimestre.

### Prazo para o envio da atividade

A entrega da atividade deverá ser realizada até o dia **29/09/2025**, às 23h59, via AVA no local apropriado para submissão.

Não serão aceitas entregas por e-mail ou torpedo, assim como após o prazo estabelecido.

### Referências

### Ferramentas e Recursos

- [Draw.io](https://draw.io/) (diagramas): <https://draw.io/>
- Figma (prototipação): <https://www.figma.com/>
- Marvel App (prototipação): <https://marvelapp.com/>
- Swagger (especificação de APIs): <https://swagger.io/>
- DbDiagram (modelagem de dados): <https://dbdiagram.io/>
- Linguagem de marcação markdown (playground): <https://dotmd-editor.vercel.app/>