



UNIVERSIDADE FEDERAL DO CEARÁ - UFC
CAMPUS CRATEÚS
CURSO DE SISTEMAS DE INFORMAÇÃO/ CIÊNCIA DA COMPUTAÇÃO

Disciplina: CRT0432 - Projeto Integrador II

Professor: Bruno Riccelli dos Santos Silva

Semestre: 2025.2

Documento Integrador

1. Introdução

Este documento tem como objetivo geral orientar o desenvolvimento do projeto integrador II para as turmas de Sistemas de Informação (SI) e Ciência da computação (CC), de forma a estabelecer os requisitos e métodos de avaliação aplicados à disciplina. Esta disciplina visa aplicar de forma prática e integrada os conhecimentos adquiridos nas disciplinas de Análise e projeto de sistemas e redes de computadores (SI) e Redes de computadores, APS, Programação Funcional, Computação Gráfica, Linguagens Formais e Autômatos (CC). Ressalta-se que é uma disciplina com carga horária de extensão, logo NECESSARIAMENTE a comunidade deve ser beneficiada com o desenvolvimento de um software e esta deve retornar com feedbacks construtivos de forma a aprimorar a aplicação.

2. Tema e Contexto

O projeto a ser desenvolvido deve abordar um problema real da sociedade e propor uma solução computacional que integre os conhecimentos das disciplinas cursadas até o momento.

3. Objetivos do Projeto

Nesta seção serão descritos os objetivos geral e específicos do projeto, os quais devem ser levados em consideração.

3.1 Objetivo Geral

Desenvolver uma aplicação que solucione o problema na comunidade, integrando conceitos e técnicas das disciplinas cursadas até o quarto período.

3.2 Objetivos Específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos são listados.

- Capacitar equipes no desenvolvimento em equipe.
- Aplicar metodologias de análise e projeto de sistemas.
- Realizar levantamento de requisitos funcionais e não funcionais.
- Modelar e implementar um banco de dados relacional.
- Projetar interfaces seguindo princípios de usabilidade e experiência do usuário.
- Elaborar o diagrama de casos de uso.
- Utilizar o conceito de redes de computadores para criação de API ou sockets, de forma a criar mecanismos de interconexão de redes e comunicação entre diferentes componentes do sistema.
- Implementar a solução com base em um sistema composto por Backend e Frontend, **não necessariamente sendo um sistema web**.

4. Requisitos

Nesta seção serão exibidos os aspectos relacionados aos requisitos que serão necessários ao longo do projeto.

4.1 Priorização de Requisitos

Para cada requisito identificado, atribuir uma das seguintes prioridades:

- **Essencial:** Imprescindível para o funcionamento básico do sistema
- **Importante:** Agrega valor significativo, mas o sistema funciona sem ele
- **Desejável:** Seria bom ter, mas não compromete o valor principal do sistema

4.2 Validação de Requisitos

Antes da entrega do Marco 1, cada equipe deve validar seus requisitos com base na lista de verificação:

- Os requisitos são claros e comprehensíveis?
- Os requisitos são testáveis/verificáveis?
- Os requisitos são viáveis com as tecnologias disponíveis?
- Os requisitos são relevantes para o problema que está sendo resolvido?
- Existe conflito entre requisitos?

4.3 Dicas para um Bom Levantamento de Requisitos

1. Pense como usuário: "O que eu esperaria desse sistema?"
2. Considere diferentes perfis de usuários e suas necessidades
3. Use verbos na voz ativa para requisitos funcionais
4. Seja específico e evite termos ambíguos/generalistas como "rápido", "amigável", "eficiente"
5. Questione sempre: "Como saberemos se esse requisito foi atendido?"

Uma das formas de se medir a qualidade de um sistema de software é pela sua utilidade. E um sistema será útil para seus usuários se atender aos requisitos definidos e se esses requisitos refletirem as necessidades dos usuários. Portanto, os requisitos devem ser expressos de uma maneira tal que eles possam ser verificados e comunicados a leitores técnicos e não técnicos. A equipe técnica (leitores técnicos) deve entender o documento de requisitos de tal forma a poder obter soluções técnicas apropriadas. Clientes (leitores não técnicos) devem entender esse documento para que possam priorizar o desenvolvimento dos requisitos, conforme as necessidades da organização em que trabalham. **Muitos sistemas foram abandonados ou nem chegaram a uso porque os membros da equipe não dispensaram tempo suficiente para compreender as necessidades do cliente em relação ao novo sistema.**

Muito importante: A menos que o sistema a ser desenvolvido seja bastante simples e estático (características raras nos sistemas atuais), é praticamente impossível pensar em todos os detalhes a princípio. Além disso, quando o sistema entrar em produção e os usuários começarem a utilizá-lo, eles descobrirão requisitos nos quais não tinham pensado inicialmente. Em resumo, os requisitos de um sistema complexo inevitavelmente mudarão durante o seu desenvolvimento.

5. Metodologia de Desenvolvimento

5.1 Formação das Equipes

As equipes serão formadas por **6 a 7** discentes, devendo ser registradas até o dia 03/10 por email: bruno.silva@crateus.ufc.br. Em seguida, as equipes deverão definir o tema do projeto **buscando problemas na comunidade de Crateús e região para solucionar**. O cronograma está definido no final do documento. **Não será possível utilizar projetos de extensão ou projetos próprios sem aplicação real.**

5.2 Processo de Desenvolvimento

O projeto será desenvolvido em iterações, com entregas incrementais:

1. **Concepção e Planejamento:**
 - Definição do problema e escopo

- Levantamento de requisitos
 - Elaboração do plano de projeto
2. **Elaboração:**
- Modelagem de dados via UML/Banco de dados
 - Prototipação das interfaces
 - Definição da arquitetura do sistema
3. **Implementação:**
- Desenvolvimento dos módulos
 - Integração dos componentes
4. **Finalização:**
- Testes com usuários
 - Ajustes e correções
 - Documentação final (Documento de transferência de conhecimento)

6. Entregas e Avaliação

Nesta seção serão definidos os marcos de entrega ao longo da disciplina.

6.1 Marco 1 (Peso 1)

- Documento de visão e escopo do projeto.
- Levantamento de requisitos funcionais e não funcionais.
- Modelo ER inicial.
- Casos de Uso.
- Protótipos de alta fidelidade.
- Apresentação oral do planejamento.

6.2 Marco 2 (Peso 1)

- Sistema implementado com todas as funcionalidades.
- Banco de dados implementado.
- Protótipos de alta fidelidade implementados.
- Demonstração das funcionalidades.
- Integração entre Frontend e Backend.
- Documentação técnica.
- Relatório de aplicação dos conceitos de cada disciplina (se houver).
- Apresentação final do projeto, com demonstração do funcionamento.

6.3 Avaliação Final - Data:

- Demonstração do sistema em funcionamento com as funcionalidades restantes
- Entrega da documentação completa

6.4 Critérios de Avaliação

Nesta seção serão apresentados os critérios de avaliação de cada marco do projeto.

6.4.1 Critérios de Avaliação do Marco 1

Critério	Descrição	Pontuação Máxima
Documento de Visão	<ul style="list-style-type: none">• Problema claramente definido• Escopo delimitado (o que será e o que NÃO será feito)• Público-alvo e beneficiários identificados• Justificativa e impacto social documentados	10 pontos
Requisitos Funcionais (RF)	<ul style="list-style-type: none">• Mínimo de 10 requisitos funcionais documentados• Cada RF deve ter: ID, descrição clara, prioridade• Priorizados como: Essencial/Importante/Desejável• Requisitos claros, testáveis e sem ambiguidades• Verbos na voz ativa (ex: "O sistema deve calcular...")• Evitar termos vagos como "rápido", "fácil", "eficiente"	10 pontos
Requisitos Não Funcionais (RNF)	<ul style="list-style-type: none">• Mínimo de 5 requisitos não funcionais• Métricas objetivas e mensuráveis• Exemplos válidos: "Tempo de resposta < 2s", "Disponibilidade de 99%"• Categorias: desempenho, segurança, usabilidade, confiabilidade	5 pontos
Modelo Entidade-Relacionamento	<ul style="list-style-type: none">• Entidades bem definidas com atributos• Relacionamentos corretos com cardinalidades (1:1, 1:N, N:N)• Chaves primárias e estrangeiras identificadas• Diagrama legível e bem formatado	8 pontos
Casos de Uso	<ul style="list-style-type: none">• Mínimo de 5 casos de uso documentados• Cada caso deve conter: nome, ator(es), pré-condições, fluxo principal• Fluxos alternativos e de exceção quando aplicável• Diagrama de casos de uso (UML) apresentado• Cobertura das principais funcionalidades do sistema	7 pontos

Protótipos de Alta Fidelidade	<ul style="list-style-type: none"> Todas as telas principais criadas Navegação funcional entre telas Interface de usuário (UI) consistente Consideração de princípios de usabilidade e UX Elementos visuais próximos ao produto final 	8 pontos
Apresentação Oral	<ul style="list-style-type: none"> Clareza na comunicação Domínio do conteúdo apresentado Capacidade de responder questionamentos Participação equilibrada dos membros 	2 pontos
TOTAL		50 pontos

6.4.2 Critérios de Avaliação do Marco 2

Critério	Descrição Detalhada	Pontuação Máxima
Funcionalidades Implementadas	<ul style="list-style-type: none"> Todos os requisitos funcionais ESSENCIAIS implementados Mínimo de 60% dos requisitos IMPORTANTES implementados Funcionalidades devem estar operacionais e testáveis Aderência aos requisitos especificados no Marco 1 	15 pontos
Frontend	<ul style="list-style-type: none"> Interface implementada conforme protótipos do Marco 1 Feedback visual para ações do usuário Tratamento de estados (loading, erro, sucesso) 	7 pontos
Integração Frontend-Backend	<ul style="list-style-type: none"> Comunicação fluida entre cliente e servidor Dados sincronizados corretamente Requisições e respostas funcionando adequadamente 	6 pontos
Banco de Dados	<ul style="list-style-type: none"> esquema implementado conforme Modelo ER Dados sendo persistidos e recuperados corretamente Integridade referencial mantida 	3 pontos

Qualidade do Código	<ul style="list-style-type: none"> • Código organizado e legível • Seguir padrões de nomenclatura consistentes • Comentários em trechos complexos • Estrutura de pastas/arquivos lógica 	4 pontos
Containerização (Docker)	<ul style="list-style-type: none"> • Aplicação completamente containerizada • Documentação de como executar os containers • Variáveis de ambiente configuradas adequadamente • Sistema roda em ambiente limpo apenas com Docker 	3 pontos
Documentação Técnica	<ul style="list-style-type: none"> • README completo e claro • Instruções de instalação e execução • Tecnologias e dependências listadas • Estrutura do projeto explicada 	8 pontos
Apresentação Final	<ul style="list-style-type: none"> • Demonstração ao vivo do sistema funcionando • Capacidade de responder questionamentos técnicos • Participação de todos os membros da equipe 	4 pontos
TOTAL		50 pontos

Nota final = (Nota do marco 1 + Nota do marco 2)/10

7. Acompanhamento e Suporte

O professor estará disponível para orientação durante o semestre através de acompanhamento quinzenal com todas as equipes, as quais deverão realizar registro em ata. Além disso, monitores das disciplinas integradas poderão ser consultados para auxílio em tópicos específicos.

8. Extensão e Impacto Social

O projeto deve considerar seu potencial impacto na comunidade local e como poderá ser aplicado para resolver problemas reais. As equipes devem identificar e documentar:

1. Potenciais usuários ou beneficiários da solução
2. Formas de aplicação prática na comunidade
3. Limitações e desafios para implantação real
4. Possibilidades de continuidade e evolução do projeto

10. Orientações Gerais

- A equipe deve ser formada por até 7 (sete) discentes.
- O tema para o projeto será decidido com algum cliente na comunidade
- O código-fonte deve ser versionado em repositório Git.
- O código fonte deve ser conteinerizado utilizando docker, visando o deploy em ambiente relevante.
- As equipes devem manter registros das reuniões e decisões tomadas. A cada reunião entre a equipe/professor deve ser gerada uma ata para o devido acompanhamento.
- Marco I: Documento de requisitos, diagramas de classe, casos de uso e protótipo elaborado no Figma com todas as telas prontas.
- Marco II: Refinamento e apresentação do sistema funcionando com backend e frontend.

11. Regras Gerais

Art. 1º - Qualquer linguagem de programação, framework, biblioteca ou plugin são permitidos para desenvolvimento do projeto, desde que não descumpram nenhuma regra específica do projeto do grupo.

Art. 2º - Problemas com os arquivos do trabalho como perda, problemas no computador, no pendrive, em alguma mídia de armazenamento ou posse dos arquivos estar com apenas um dos membros, não será motivo para evitar penalização por atraso na entrega da atividade. O grupo é responsável por manter cópias dos arquivos para evitar a perda ou impossibilidade de entrega.

Art. 3º - A última versão do arquivo executável do projeto deve estar no repositório de controle de versão usado pelo grupo na data de entrega do marco.

Art. 4º - No Marco I o grupo apresentará a versão do seu projeto com o que foi solicitado no plano de curso da disciplina.

Art. 5º - No Marco II o grupo apresentará a versão final do seu projeto, englobando o que foi solicitado no Marco I. Para ser aprovado, o grupo deve ter cumprido os critérios de avaliação do projeto.

Art. 6º - Se alguma das disciplinas envolvidas no projeto estiver com o conteúdo atrasado, as funcionalidades relacionadas à disciplina poderão ser reduzidas, a critério do professor.

Art. 7º - Caso a versão final não seja aprovada no Marco II, o grupo estará de Prova Final e deverá fazer uma nova apresentação da versão no dia estipulado.

Art. 8º - Cada equipe terá 20 minutos para apresentação do marco 1 e 30 minutos para a apresentação do marco 2

Art. 9º - É proibido que o projeto aborde os seguintes tópicos:

- Referências a drogas ilícitas, tabaco, marcas de produtos e temas questionáveis (ao critério do professor);
- Referência ou apologia a violência ou crimes (ao critério do professor);
- Material relacionado a atividades ilegais como jogos de azar, empresas, serviços ou produtos que promovem ou estimulam tais jogos;
- Material difamatório, obsceno, profano, vulgar, repulsivo, ofensivo ou sugestivo que descreva ou represente funções fisiológicas, enfermidades ou qualquer assunto que uma pessoa de bom senso julgaria socialmente inaceitável (ao critério do professor);
- Publicidade para produtos ou sites pornográficos;
- Materiais que contenham marcas registradas, conteúdo protegido por direitos autorais ou outras formas de propriedade intelectual sem o consentimento do proprietário, ou qualquer elemento que possa submeter o professor ou a universidade em processos por infração ou apropriação indevida;
- Material que possa ofender qualquer grupo, aluno, pessoa, entidade ou produto.

Art. 10º - O desconhecimento da existência do trabalho ou datas das atividades do trabalho não serão aceitos como justificativa para não realização ou atraso da atividade. É obrigação do grupo buscar informações no SIGAA, com o professor ou demais alunos sobre atividades da disciplina.

Art. 11º - A não compreensão da descrição e realização da atividade não será aceita como justificativa para realização incompleta ou incorreta das atividades do trabalho. É obrigação do aluno sanar dúvidas em relação às atividades.

Art. 12º - A descrição dos projetos e suas exigências podem sofrer alterações no decorrer da disciplina, as quais serão comunicadas ao grupo.

Art. 13º - Será criado um grupo no google entre os participantes e o professor para o devido acompanhamento.

12. SUGESTÃO de funções no projeto

Levando em consideração um projeto de desenvolvimento, pode-se pensar nos perfis que comumente são presentes neste contexto. Portanto, os seguintes papéis auxiliam na divisão de tarefas no grupo:

- **Analista de Requisitos e Designer:** conversa com o cliente, levanta e documenta necessidades. Além disso, cria os protótipos e cuida da usabilidade do sistema. Um integrante.
- **Desenvolvedor Backend** – implementa a lógica do servidor, APIs e banco de dados. Dois integrantes.
- **Desenvolvedor Frontend** – constrói a interface do usuário e integra com a API. Dois integrantes.
- **Tester/QA** – planeja e executa testes para garantir qualidade e funcionamento. Um integrante.

Ressalto que a denominação frontend e backend não estão necessariamente ligadas a desenvolvimento web. Uma aplicação local também pode ser dividida desta forma.

12. Cronograma

Atividade	Data
Definição das equipes	Até dia 03/10
Definição do tema de cada equipe	Até dia 11/10
Levantamento inicial de requisitos Protótipos de baixa fidelidade Início do modelo ER preliminar	Até 02/11
Refinamento dos requisitos com cliente/professor Protótipos de alta fidelidade Modelo ER revisado Diagrama de casos de uso inicial	Até 16/11
Marco I - Apresentação de 20 minutos por equipe	Semana do dia 17/11
Marco II	Semana do dia 19/01/2026
Acompanhamento das atividades	Semanal ou quinzenal, a combinar com cada equipe
Avaliação Final	23/01/2026

13. FAQ

1. Quantas pessoas devem compor as equipes?

R - As equipes devem ter entre 6 e 7 integrantes.

2. Podemos escolher qualquer tema para o projeto?

R - O tema precisa abordar um problema real da comunidade de Crateús e região, com um cliente ou beneficiário definido. Projetos sem aplicação prática não serão aceitos.

3. O projeto precisa ser inédito?

R - Não necessariamente. A equipe pode propor melhorias para uma solução já existente, desde que traga inovação ou adaptação ao contexto local.

4. Todos os requisitos elicitados precisam ser implementados até o final?

R- Não necessariamente. Os requisitos implementáveis devem ser combinados junto do cliente e professor

5. O sistema precisa ser web obrigatoricamente?

R - Não. Pode ser web, mobile ou desktop.

6. Há restrições sobre linguagens e frameworks?

R- Não. A equipe pode escolher qualquer linguagem, framework ou biblioteca, desde que atenda aos requisitos e mantenha versionamento no Git.

7. Como devem ser feitos os protótipos?

R- Os protótipos de alta fidelidade devem ser entregues preferencialmente no Figma. Outras ferramentas podem ser usadas, desde que tenham o mesmo nível de detalhamento.

8. A apresentação final terá perguntas?

R- Sim. Além de apresentar, a equipe deverá responder questionamentos

9. Professor, a disciplina é um TCC?

R - Não.

10. É muita coisa para um PI?

R - Para 6 integrantes, não. Porém requer planejamento e trabalho em equipe. Esta disciplina também serve para lidar com soft skills, as quais serão utilizadas no mercado de trabalho.

11. O professor ficará cobrando as atividades?

R - Serão realizadas reuniões quinzenais, as quais três perguntas devem ser respondidas:

- O que foi feito?
- Houve algum impedimento?
- O que será feito para a próxima reunião?

12. Posso ficar sozinho na equipe ?

R - Não.

Anexo 1 - Ata de acompanhamento quinzenal

Equipe: _____

Data: ____ / ____ / ____

Participantes: _____

1. Progresso desde o último encontro

Resuma as atividades realizadas por integrante.

2. Dificuldades encontradas

Listar os principais obstáculos técnicos, de organização ou de comunicação.)

3. Feedback do Professor

4. Metas para os próximos 15 dias