



Instituto Infnet

Desenvolvimento Front-end com Frameworks
[25E4-26E1]

Projeto de Bloco

2025T4

Grupo:

Arthur Teixeira Barcellos

Geusiene Fernanda Rodrigues da Silva

Lucas Amorim Porciuncula

Wellington Albuquerque Falcão

Professor: Tiago Cariolano de Souza Xavier

Sumário

Sumário.....	2
1. Definição de Projeto.....	3
2. Definição de Escopo e Objetivos.....	3
2.1 Versão TP1 – Escopo Inicial.....	3
2.2 Atualização da Definição de Escopo e Objetivos (TP2).....	4
3. Gestão Ágil do Projeto.....	5
4. Artefatos, Papéis e Eventos do Scrum.....	8
5. Criação de Histórias de Usuário.....	10
5.1 História de Usuário 1 e História de Usuário 2 (TP1).....	11
5.2 Atualização da Seção Criação de História de Usuários (TP2).....	13
5.3 Atualização da Seção Criação de História de Usuários (TP3).....	15
6. Planejamento do Projeto de Desenvolvimento de Front-end.....	16
7. Interatividade na Aplicação Web/Mobile.....	18
8. Framework.....	20
8.1 Versão TP1:.....	20
8.2 Atualização da Seção Framework (TP2).....	21
8.2 8.2 Atualização da Seção Framework (TP3).....	22

1. Definição de Projeto

Descreva qual a aplicação foi escolhida pelo professor para ser desenvolvida.

1. Definição de Projeto. Está ok.

O professor dividiu a turma em grupos e nos deu a oportunidade de optar entre dois tipos de projetos: Desenvolver uma aplicação web de Leilões ou desenvolver uma aplicação de compra e venda de Bitcoins, ambos utilizando os aprendizados em aula com React, Mobile-First e metodologias Ágeis . Nosso grupo escolheu seguir com o projeto de Leilões.

2. Definição de Escopo e Objetivos

Qual é o objetivo principal da aplicação web que desenvolveremos, e quais funcionalidades específicas devem ser incluídas para atender às necessidades do escopo definido pelo professor?

2.1 Versão TP1 – Escopo Inicial

O objetivo principal é criar uma Aplicação Web (e Mobile) de Leilões de Imóveis que permita a administração de leilões, a visualização de lotes, e a realização de lances em tempo real pelos usuários cadastrados. O foco é garantir a transparência, a segurança e a interatividade da plataforma.

Como funcionalidades específicas destacamos:

- CRUD - Cadastrar e consultar os lances de imóveis em tempo real. Fazer lances, verificar se o lance foi contemplado, demonstrar os horários dos leilões.
- Cadastrar Leilão;
- Histórico de Lances
- Cadastro de usuários por tipo - Arremate

Feedback TP1:

Definição de Escopo e Objetivos. Está ok, no entanto o objetivo não menciona explicitamente responsividade, versão mobile ou gestos, que fazem parte dos critérios obrigatórios do projeto e que deveriam aparecer já no escopo. As funcionalidades são apresentadas como listas estáticas, porém falta contextualizar como elas atendem ao objetivo proposto.

2.2 Atualização da Definição de Escopo e Objetivos (TP2)**Objetivo da Sprint / TP2:**

Com base no feedback do TP1, o objetivo foi reformulado para incluir explicitamente responsividade, abordagem Mobile-First e gestos, além de detalhar a simulação de funcionalidades.

Objetivo atualizado (TP2):

O objetivo principal passou a ser desenvolver uma aplicação Web do tipo single-page (SPA), utilizando React / React Native, com abordagem Mobile-First, de leilões de imóveis que permita a administração de leilões, a visualização de lotes e a realização de lances utilizando dados *mockados* e usuários simulados.

Ressalta-se que, nesta etapa2 do projeto (TP2), as funcionalidades são simuladas, não sendo implementadas operações completas de CRUD nem integração com banco de dados, conforme o escopo definido pelo professor.

A aplicação será responsiva, seguindo o conceito mobile first: celulares serão a base do design, enquanto tablets, desktops e notebooks devido ao maior espaço de tela terão funcionalidades mais amplas e visíveis, respeitando a usabilidade de cada dispositivo

Smartphones e Tablets

Devido ao tamanho reduzido da tela, a usabilidade é focada na apresentação dos principais detalhes do imóvel, reduzindo ao máximo a quantidade de informações exibidas e aplicando

padrões consolidados de navegação, como:

- Movimentos de pinça para zoom;
- Navegação simples e objetiva entre telas ou seções;
- Deslizar (slide) para alternância de imagens.
- Priorização do conteúdo essencial para o usuário.

Em dispositivos com telas maiores, como tablets, será possível exibir mais informações simultaneamente, sem comprometer a clareza e a usabilidade da interface.

3. Gestão Ágil do Projeto

Como organizaremos o trabalho seguindo os preceitos do Agile Scrum? Quais serão nossos sprints, como definiremos as tarefas, e como as atribuiremos para assegurar o desenvolvimento efetivo e o cumprimento dos prazos? Mostre uma primeira versão do cronograma de desenvolvimento do projeto.

3. Gestão Ágil do Projeto. Incompleto. Não explica como as tarefas serão definidas, como o enunciado exige. Não explica como as tarefas serão atribuídas. Sprint semanal é incompatível com o cronograma gigante mostrado em semanas anteriores (inconsistência temporal).

Definição das Tarefas

As tarefas do projeto serão definidas a partir do Product Backlog, que será elaborado e priorizado pelo Product Owner. Esse backlog conterá:

- Funcionalidades principais da aplicação, tais como visualização de lotes, simulação de lances e navegação entre telas;
- User Stories (funcionalidades específicas, escritas do ponto de vista do usuário);
- Critérios de aceitação para cada história;
- Estimativa de complexidade (dias ou pontos).

O processo ocorrerá da seguinte forma:

Refinamento do Backlog

O refinamento será realizado semanalmente, ocasião em que:

- Cada User Story será detalhada;
- As histórias serão decompostas em tarefas menores, voltadas principalmente à construção de componentes e estilos em React / React Native;
- Serão identificadas dependências;
- A equipe colaborará na estimativa do esforço necessário para cada tarefa;
- Os critérios de aceitação serão validados para garantir alinhamento com o escopo de simulação definido a cada TP.

Uso do Planning Poker na Estimativa

Durante o refinamento, será adotada a técnica **Planning Poker**, a fim de garantir uma **estimativa colaborativa**, precisa e consensual. O processo funcionará da seguinte forma:

- Cada desenvolvedor receberá um conjunto de cartas de estimativa baseado na escala de Fibonacci (1, 2, 3, 5, 8, 13, 21...);
- O Product Owner apresentará a User Story e esclarecerá dúvidas antes da votação;
- A equipe, simultaneamente, revelará sua estimativa utilizando as cartas;
- Caso haja divergências significativas, o time discutirá justificativas técnicas;
- Uma nova rodada de votação ocorrerá até que se alcance consenso ou um valor médio acordado.

Esse método garantirá **maior participação da equipe, redução de vieses individuais, maior precisão das estimativas e construção de um entendimento compartilhado** sobre a complexidade de cada tarefa.

Planejamento do Sprint

No início de cada ciclo de duas semanas, o time selecionará coletivamente as histórias que serão desenvolvidas no Sprint, considerando:

- prioridade estabelecida pelo Product Owner;
- capacidade de trabalho da equipe;
- complexidade estimada no Planning Poker;
- dependências técnicas e riscos.

Assim, cada funcionalidade presente no cronograma será decomposta em tarefas realizáveis dentro do Sprint, sempre baseada na estimativa colaborativa definida pelo time.

Atribuição das Tarefas

Durante o Sprint Planning, as tarefas serão distribuídas por auto-organização entre os integrantes do time, conforme recomendado pelo framework Scrum. Cada membro da equipe selecionará as atividades nas quais possui maior domínio técnico ou disponibilidade.

O Scrum Master atuará para:

- garantir o equilíbrio da carga de trabalho;
- apoiar na remoção de impedimentos;
- assegurar que o planejamento seja realista e compatível com o tempo disponível do Sprint.

Quando necessário, tarefas relacionadas à integração de componentes ou fluxos de navegação poderão ser desenvolvidas em pares, favorecendo a colaboração e a qualidade do resultado final.

Estrutura de Sprints

A execução das fases ocorrerá dentro de Sprints de duas semanas, garantindo coerência temporal:

- Fase 1 (4–6 semanas) → 2 a 3 Sprints
- Fase 2 (6–8 semanas) → 3 a 4 Sprints
- Fase 3 (4–6 semanas) → 2 a 3 Sprints
- Fase 4 (Contínuo) → Sprints contínuos para ajustes e melhorias

Dessa forma, todo o cronograma permanecerá consistente com a filosofia ágil baseada em iterações, atendendo ao apontamento do professor.

Detalhamento e gráfico Gantt para melhor compreensão do planejamento após feedback do TP1:

Cronograma Geral

Fase 1: Configuração e MVP Web (4–6 semanas)

Objetivo: estabelecer a base técnica e entregar o MVP da aplicação Web para administradores.

Entregáveis: arquitetura definida, cadastro de usuários, módulo de cadastro de leilões e visualização de lotes.

Fase 2: Interatividade em Tempo Real (6–8 semanas)

Objetivo: implementar a funcionalidade central de lances em tempo real.

Entregáveis: Visualizar lances, histórico, mecanismo de arremate e notificações.

Fase 3: Refinamento e Mobile (4–6 semanas)

Objetivo: realizar aprimoramentos, testes de segurança e desenvolver a aplicação mobile.

Entregáveis: aplicativo mobile funcional, otimizações de desempenho, testes de carga e documentação.

Fase 4: Lançamento e Pós-Lançamento (Contínuo)

Objetivo: implantação, monitoramento e melhorias.

Entregáveis: deploy, monitoramento ativo, correções e backlog de evolução.

Roadmap

		2025												2026													
Projeto	▼	Setembro				Outubro				Novembro				Dezembro				Janeiro				Fevereiro				Março	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2				
Fase 1: Configuração e MVP Web																											
Fase 2: Interatividade em Tempo Real																											
Fase 3: Refinamento e Mobile																											
Fase 4: Lançamento e Pós-Lançamento																											

Roadmap com backlog

		2025												2026													
Projeto	▼	Setembro				Outubro				Novembro				Dezembro				Janeiro				Fevereiro				Março	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2				
Fase 1: Configuração e MVP Web																											
Definição da Arquitetura																											
Dev. Cadastro de Usuários																											
Dev. Módulo de cadastro de leilões																											
Dev. Visualizações de Lotes																											
Fase 2: Interatividade em Tempo Real																											
Dev. CRUD Lances																											
Dev. Histórico																											
Dev. Mecanismo de arremate																											
Dev. Notificações																											
Fase 3: Refinamento e Mobile																											
Dev. Aplicativo mobile funcional																											
Otimizações de desempenho																											
Testes de carga																											
Documentação																											
Fase 4: Lançamento e Pós-Lançamento																											
Deploy																											
Monitoramento Ativo																											
Correções																											
Backlog de evolução																											

4.Artefatos, Papéis e Eventos do Scrum

Como você planeja incorporar os artefatos, papéis, e eventos do Scrum no desenvolvimento da aplicação web? Descreva como cada elemento será utilizado para garantir um fluxo de trabalho ágil e eficiente.

4. Artefatos, Papéis e Eventos do Scrum. Está ok, no entanto o Scrum Master e o Product Owner acumulados na mesma pessoa é aceitável em grupos pequenos, mas isso deveria ser justificado. Não há explicação de como cada artefato será usado no projeto específico, só explicações genéricas. “Weekly Scrum” está errado — o nome correto é Daily Scrum.

Alterações realizadas na tabela abaixo:

Elemento Scrum	Descrição e Utilização no Projeto	Responsável
Papel: Product Owner (PO)	Responsável pela visão do produto. Definirá e priorizará o Product Backlog (lista de HUs), garantindo que as funcionalidades atendam aos requisitos do professor e do mercado (ex: foco na interatividade).	Arthur
Papel: Scrum Master (SM)	Facilitador. Garantirá que a equipe siga os preceitos do Scrum, removerá impedimentos (ex: problemas com API, bloqueios técnicos) e protegerá a equipe de interrupções.	Geusiene
Papel: Time de Desenvolvimento	Responsável pela entrega do produto. Criará as Histórias de Usuário, estimará o trabalho e construirá a aplicação (Front-end com React/React Native, Back-end).	Lucas/Wellington
Artefato: Product Backlog	A lista única e priorizada de todas as funcionalidades do leilão (HUs, requisitos). Será revisado e atualizado continuamente (Refinamento de Backlog).	Arthur/Geusi
Artefato: Sprint Backlog	Conjunto de HUs e tarefas selecionadas pelo Time e PO para o Sprint atual. É o plano do Time para as próximas 2 semanas.	Todos
Artefato: Incremento	A soma de todos os itens do Sprint Backlog completados durante um Sprint, funcionando e testado. Será a versão funcional da aplicação de leilão ao final de cada Sprint.	Todos
Evento: Sprint Planning	O Time e PO se reúnem para selecionar HUs do Product Backlog e planejar como serão construídas na Sprint (definição do Sprint Backlog).	Todos
Evento: Daily Checkpoint	Reunião diária de 15 minutos para inspecionar o progresso em direção ao Objetivo da Sprint e ajustar o plano. Ocorre todos os dias às 09:00.	Todos
Evento: Sprint Review	Ao final da Sprint, o Incremento é demonstrado ao professor/stakeholders para coletar feedback sobre as funcionalidades implementadas (ex: funcionalidade de lance).	Todos
Evento: Sprint Retrospective	Reunião interna para o Time inspecionar a si mesmo e criar um plano para melhorias no próximo Sprint (processo, ferramentas, colaboração).	Todos

5.Criação de Histórias de Usuário

Como você irá estruturar as histórias de usuário para este projeto? Crie duas histórias de usuário que capturam requisitos funcionais e não funcionais, incluindo responsividade,

consumo de dados de uma fonte externa, e funcionalidades específicas para a versão mobile que utilizem gestos do usuário.

5.1 História de Usuário 1 e História de Usuário 2 (TP1)

HISTÓRIA 1

Título: Busca e Cadastro de Imóvel via CEP

Como: Usuário Administrador

Eu quero: Poder cadastrar um novo imóvel no sistema informando apenas o CEP

Para que: Os dados de endereço sejam preenchidos automaticamente e eu ganhe agilidade no cadastro

Critérios de Aceitação:

- Deve consumir a API ViaCEP ou similar para buscar dados do endereço
- Deve preencher automaticamente os campos: logradouro, bairro, cidade e estado
- Deve funcionar tanto na versão web quanto mobile
- Em caso de CEP inválido, deve exibir mensagem de erro claramente
- O layout deve ser responsivo, adaptando-se a diferentes tamanhos de tela

HISTÓRIA 2:

Título: Navegação por Gestos em Lotes de Imóveis

Como: Usuário Arrematante

Eu quero: Poder navegar entre os lotes de imóveis usando gestos de deslize (swipe)

Para que: Tenha uma experiência mais fluida e intuitiva ao visualizar os imóveis em leilão

Critérios de Aceitação:

- Deve permitir navegar entre imóveis deslizando horizontalmente na tela
- Deve exibir indicador visual de progresso (ex: "2 de 15")
- O gesto deve ser suave e responsivo, sem travamentos
- Deve funcionar offline para imóveis já carregados
- Deve manter a sincronização em tempo real quando online

Requisitos Funcionais (O QUE o sistema faz)

1. Gestão de Catálogo de Imóveis

- Listar todos os imóveis em leilão
- Visualizar detalhes de um imóvel específico
- Cadastrar imóvel através do CEP (via API)
- Filtrar imóveis por localização (estado/cidade)

2. Gestão de Usuários

- Cadastro básico de usuários
- Login/logout simplificado

3. Sistema de Lances

- Dar lance em imóvel disponível
- Validação de valor mínimo (acima do lance atual)
- Verificação de leilão ativo/encerrado
- Atualização em tempo real do lance atual

4. Gestão de Leilões

- Controle de datas de início/término
- Status do leilão (em andamento/encerrado/futuro)

Funcionalidades Não-Funcionais (COMO o sistema funciona)**1. Desempenho**

- Tempo de resposta < 3 segundos para operações principais
- Suporte a até 50 usuários simultâneos

2. Segurança

- Validação de dados de entrada

3. Usabilidade

- Interface intuitiva e minimalista
- Mensagens de erro claras para o usuário

4. Confiabilidade

- Disponibilidade de 98% durante horário comercial
- Prevenção de perda de dados críticos (lances)

5. Arquitetura

- Armazenamento simples (banco em memória)

Feedback TP1:

Criação de Histórias de Usuário. Está bom, no entanto faltou incluir critérios de testes mais formais ou mensuráveis.

5.2 Atualização da Seção Criação de História de Usuários (TP2)

Baseado no Feedback feito pelo professor do TP1, as seguintes atualizações foram realizadas:

- Foram incluídos testes incluídos em cada história de usuário.
- Foram incluídos critérios de aceite e cenários de testes formais

HISTÓRIA 1: Busca e Cadastro de Imóvel via CEP (TP2)

Título: Cadastro Automatizado de Endereço via Integração de API

Como: Usuário Administrador

Eu quero: Poder cadastrar um novo imóvel informando apenas o CEP e ter os campos preenchidos automaticamente

Para que: Eu elimine erros de digitação e reduza o tempo de cadastro de cada imóvel.

Critérios de Aceite e Cenários de Teste Formais**1. Sucesso na Integração (Teste Funcional)**

- **Dado** que o administrador está na tela de "Novo Imóvel";
- **Quando** ele digitar um CEP válido (ex: 24220-900) e o foco sair do campo (evento *onBlur*) ou clicar em "Buscar";
- **Então** o sistema deve simular a busca de um endereço a partir de uma fonte simulada
- **E** o tempo de resposta deve ser **inferior a 2 segundos** (Requisito Não-Funcional de Desempenho);
- **E** os campos Logradouro, Bairro, Cidade e Estado devem ser preenchidos com os dados retornados;
- **E** o cursor deve ser posicionado automaticamente no campo Número.

2. Tratamento de Erros e Validação (Teste de Exceção)

- **Dado** que o administrador inseriu um CEP inexistente ou com formato inválido (ex: 00000-000 ou abcde-fgh);

- **Quando** a validação for acionada;
- **Então** a borda do campo CEP deve ficar vermelha (#FF0000);
- E deve ser exibida a mensagem exata: *"CEP não encontrado ou formato inválido."*;
- E o botão de ação do formulário exibido como "Salvar" deve permanecer desabilitado até a correção.

3. Responsividade e Layout (Teste de Interface)

- **Dado** que o sistema é acessado via dispositivo móvel (viewport largura < 768px);
- **Então** os campos de endereço (Logradouro, Número, Bairro) devem se reorganizar para uma coluna única (stack vertical);
- E os campos devem manter altura mínima de **44px** (padrão de toque mobile) para garantir usabilidade.

HISTÓRIA 2: Navegação por Gestos em Lotes de Imóveis (TP2)

Título: Navegação Mobile por Gestos (Swipe)

Como: Usuário Arrematante (Mobile)

Eu quero: Navegar entre os detalhes dos lotes deslizando o dedo na tela

Para que: A experiência de visualização em dispositivos móveis seja nativa, fluida e ágil.

Critérios de Aceite e Cenários de Teste Formais

1. Mecânica do Gesto (Teste de Usabilidade)

- **Dado** que o usuário está visualizando os detalhes do "Lote A" em um dispositivo touch;
- **Quando** o usuário realizar um movimento de deslize horizontal (*swipe*) da direita para a esquerda cobrindo pelo menos **15% da largura da tela**;
- **Então** o sistema deve transicionar para o "Lote B" (próximo item);
- E a transição deve ocorrer em menos de **300ms** (animação suave sem "jank");
- E o contador de paginação deve atualizar de "1 de X" para "2 de X".

2. Comportamento Offline (Teste de Confiabilidade)

- **Dado** que o usuário carregou a lista de leilões e perdeu a conexão com a internet;
- **Quando** ele tentar deslizar para o próximo imóvel já carregado em cache (memória);
- **Então** o conteúdo (foto principal, título e valor) deve ser exibido normalmente;

- **Mas** o botão de "Dar Lance" deve estar desabilitado ou exibir alerta: "Conexão necessária para lances".

3. Limites da Navegação (Teste de Fronteira)

- **Dado** que o usuário está no **último** imóvel da lista;
- **Quando** ele tentar deslizar para a esquerda (tentar ir para o próximo);
- **Então** deve haver uma resistência elástica visual (*rubber band effect*) indicando o fim da lista;
- **E** não deve ocorrer navegação ou erro de aplicação (crash);

Feedback TP2:

Sobre O aluno atualizou as histórias de usuários apresentadas no TP1 com base nos feedbacks fornecidos pelo professor". Embora o trabalho contenha histórias de usuário bem detalhadas, com critérios de aceite formais e preocupação com responsividade e gestos mobile, não há qualquer menção explícita a ajustes realizados a partir de feedbacks do professor sobre o TP1. É preciso indicar as histórias anteriores, as novas histórias e as alterações que foram realizadas.

5.3 Atualização da Seção Criação de História de Usuários (TP3)

Com base no feedback do TP2, esta seção passou a evidenciar explicitamente a evolução das histórias de usuário ao longo das entregas.

No TP1, as histórias apresentavam critérios de aceitação descritivos e pouco mensuráveis.

No TP2, foram incorporados:

- Critérios formais no padrão Dado/Quando/Então;
- Métricas objetivas de desempenho (tempo de resposta < 2s);
- Parâmetros técnicos de responsividade (viewport < 768px);
- Requisitos explícitos de usabilidade mobile (altura mínima de 44px);
- Testes de exceção e fronteira.

No TP3, além da formalização anterior, foram adicionados:

- Quadro comparativo evolutivo das histórias;

- Identificação explícita das alterações realizadas em cada versão;
- Análise do refinamento incremental baseado nos feedbacks docentes.

Quadro Evolutivo das Histórias de Usuário:

História	TP1	TP2	TP3	Alterações realizadas
HU01 – Cadastro via CEP	Manual, critérios implícitos, sem responsividade	Automático, integração com API, critérios formais, Mobile-First	Feedback visual, validação aprimorada, animações e performance simulada	Evolução de requisitos descritivos para requisitos mensuráveis com validação formal.
HU02 – Navegação por Gestos	Clique ou scroll simples	Swipe, responsivo, offline, indicador de progresso	Ajustes em gestos, animação mais fluida, sincronização offline refinada	Evidenciamento de melhorias e refinamentos realizados

6.Planejamento do Projeto de Desenvolvimento de Front-end

Quais são os passos iniciais e os marcos principais no planejamento do nosso projeto de desenvolvimento de front-end?

6. Planejamento do Projeto de Front-end. Não há entregáveis, nem linha do tempo, nem milestones como protótipo, construção de componentes, testes, integração com API, refino de tarefas. Texto não possui aplicação prática ao projeto.

Detalhamento e gráfico Gantt para melhor compreensão do planejamento após feedback do TP1:

Cronograma Geral

Fase 1: Configuração e MVP Web (4–6 semanas)

Objetivo: estabelecer a base técnica e entregar o MVP da aplicação Web para administradores.

Entregáveis: arquitetura definida, cadastro de usuários, módulo de cadastro de leilões e visualização de lotes.

Fase 2: Interatividade em Tempo Real (6–8 semanas)

Objetivo: implementar a funcionalidade central de lances em tempo real.

Entregáveis: Visualizar lances, histórico, mecanismo de arremate e notificações.

Fase 3: Refinamento e Mobile (4–6 semanas)

Objetivo: realizar aprimoramentos, testes de segurança e desenvolver a aplicação mobile.

Entregáveis: aplicativo mobile funcional, otimizações de desempenho, testes de carga e documentação.

Fase 4: Lançamento e Pós-Lançamento (Contínuo)

Objetivo: implantação, monitoramento e melhorias.

Entregáveis: deploy, monitoramento ativo, correções e backlog de evolução.

Roadmap

Projeto		2025																2026									
		Setembro				Outubro				Novembro				Dezembro				Janeiro				Fevereiro				Março	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Fase 1: Configuração e MVP Web																											
Fase 2: Interatividade em Tempo Real																											
Fase 3: Refinamento e Mobile																											
Fase 4: Lançamento e Pós-Lançamento																											

Roadmap com backlog

		2025																2026									
		Setembro				Outubro				Novembro				Dezembro				Janeiro				Fevereiro				Março	
Projeto		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Fase 1: Configuração e MVP Web																											
Definição da Arquitetura																											
Dev. Cadastro de Usuários																											
Dev. Módulo de cadastro de leilões																											
Dev. Visualizações de Lotes																											
Fase 2: Interatividade em Tempo Real																											
Dev. CRUD Lances																											
Dev. Histórico																											
Dev. Mecanismo de arremate																											
Dev. Notificações																											
Fase 3: Refinamento e Mobile																											
Dev. Aplicativo mobile funcional																											
Otimizações de desempenho																											
Testes de carga																											
Documentação																											
Fase 4: Lançamento e Pós-Lançamento																											
Deploy																											
Monitoramento Ativo																											
Correções																											
Backlog de evolução																											

7. Interatividade na Aplicação Web/Mobile

Quais serão as interações básicas disponíveis na nossa aplicação Web/Mobile? Como elas serão planejadas para garantir uma experiência do usuário fluida e intuitiva?

7. Interatividade na Aplicação Web/Mobile. Não lista as interações básicas (clique, navegação, filtros, abrir detalhes). Não responde “como elas serão planejadas para garantir experiência fluida”. O texto apresenta apenas 3 frases desconexas. Não explica transições, gestos mobile, etc. Não conecta com as histórias de usuário criadas.

Trecho totalmente refeito para atender ao feedback do TP1:

Catálogo de Interações Básicas

As interações fundamentais do sistema foram desenhadas seguindo padrões de UI modernos para reduzir a curva de aprendizado:

- **Navegação e Drill-down:** O usuário poderá clicar/tocar no card de qualquer imóvel na listagem para abrir a visualização detalhada (Drill-down). Na versão mobile, isso ocorrerá em uma transição de tela cheia; na web, via navegação convencional.

- **Filtros e Busca:** A barra de busca será persistente no topo. Filtros avançados (localização, valor) abrirão em janelas modais (mobile) ou sidebar lateral (desktop) para não obstruir a vista dos produtos.
- **Ação de Lance (Call to Action):** O botão de "Dar Lance" terá destaque visual (cor primária e elevação) e estará fixado no rodapé da tela em dispositivos móveis (*Sticky Footer*) para estar sempre acessível durante a rolagem.
- **Formulários Inteligentes (Conexão com História 1):** Conforme definido na história de *Busca e Cadastro via CEP*, os campos de entrada possuirão validação *onBlur* (ao sair do campo) e preenchimento automático, minimizando a digitação do usuário e prevenindo erros antes do envio do formulário.

Planejamento de Fluidez e Feedback Visual

Para garantir a percepção de performance e fluidez solicitada, utilizaremos as seguintes estratégias:

- **Feedback Imediato (Micro-interações):** Toda ação do usuário terá uma resposta visual imediata.
 - *Exemplo:* Ao clicar em "Dar Lance", o botão mudará de estado (loading) instantaneamente enquanto a requisição é processada.
- **Skeleton Screens (Telas de Esqueleto):** Em vez de usar apenas um "girador" (spinner) de carregamento, o sistema exibirá a estrutura cinza do layout (cards, textos) enquanto os dados da API são baixados. Isso reduz a ansiedade do usuário e aumenta a sensação de velocidade em conexões móveis lentas.
- **Notificações Não-Obstrutivas (Toasts):** Mensagens de sucesso (ex: "Lance registrado") ou erro (ex: "CEP inválido") aparecerão como *Toasts* (pequenos cards flutuantes) no topo ou base da tela, desaparecendo automaticamente após 3 segundos, sem bloquear a navegação.

Especificidades Mobile, Gestos e Transições

A versão mobile não será apenas uma adaptação visual, mas comportamental, aproveitando as capacidades nativas de toque:

- **Navegação por Gestos (Conexão com História 2):** Conforme detalhado na história de *Navegação por Gestos*, implementaremos a funcionalidade de **Swipe (Deslize Horizontal)** na tela de detalhes.
 - *Interação:* Deslizar para a esquerda carrega o próximo lote; para a direita, o anterior.
 - *Transição:* Utilizaremos animações de "slide" aceleradas por hardware (CSS Transform) para garantir 60fps, evitando travamentos.
- **Áreas de Toque (Touch Targets):** Todos os elementos interativos (botões, links, ícones) terão uma área de toque mínima de **44x44 pixels** para evitar toques acidentais, essencial para a usabilidade em telas pequenas.

Otimização no Consumo de Dados

Para suportar conexões móveis instáveis (3G/4G):

- **Paginação e Lazy Loading:** A listagem de leilões não carregará todos os itens de uma vez. Carregaremos em blocos (ex: 10 itens) conforme o usuário rola a tela (Scroll Infinito).
- **Cache de Requisições:** Dados estáticos (como endereços retornados pela busca de CEP) serão armazenados temporariamente no navegador para evitar chamadas repetitivas à API.

8.Framework

Explique como os frameworks ReactJS e React Native irão ajudar a cumprir os requisitos descritos anteriormente.

8.1 Versão TP1:

Estes frameworks serão fundamentais para acelerar o desenvolvimento, alinhando-se às metodologias ágeis e aos requisitos do projeto. Eles permitirão a criação de uma interface de usuário dinâmica, com atualizações em tempo real e renderização no lado do cliente, o que garante uma interação mais rápida e responsiva. A arquitetura adotada assegura uma comunicação eficiente entre o front-end e a API, reduzindo a latência e tornando o processo de dar um lance quase instantâneo. O design seguirá o conceito "Mobile First", priorizando a

experiência em smartphones e tablets. Dessa forma, a usabilidade e a agilidade serão máximas no dispositivo onde o usuário mais precisa de velocidade do arremate.

Feedback TP1:

8. Framework. Está ok, no entanto muitas explicações são genéricas ("React é rápido", "interface dinâmica"). O texto não relaciona os frameworks aos requisitos obrigatórios do enunciado (API externa, mobile-gestos, responsividade). Nada é exemplificado no contexto do projeto de leilões.

8.2 Atualização da Seção Framework (TP2)

Com base no feedback recebido no TP1, o texto foi reformulado no TP2 para detalhar tecnicamente como os frameworks atendem aos requisitos funcionais e não-funcionais do sistema.

Para garantir maior celeridade e melhor aplicação dos padrões de desenvolvimento, esses frameworks são considerados boas alternativas para o front-end, também por conta da fácil manutenção e escalabilidade. A componentização permite um constante aprimoramento e reaproveitamento de funcionalidades, além de possibilitar correções mais precisas. Somado a isso, o React se destaca pela agilidade na elaboração de interfaces responsivas, adaptando-se com eficiência aos diferentes tamanhos de tela necessários para o projeto e proporcionando uma experiência fluida em dispositivos móveis, tablets e desktops.

A integração com a API externa de leilões de imóveis possibilita, por meio do React, a atualização em tempo real da interface sem a necessidade de uma interação do usuário ou atualização manual da tela, o que poderia gerar atrasos ou erros de carregamento.

O React Native representa um passo essencial dentro do conceito de mobile-first, pois permite a elaboração de uma aplicação nativa compatível com as plataformas iOS e Android, reduzindo significativamente o custo e o tempo de desenvolvimento. Além disso, como ReactJS e React Native compartilham a mesma base conceitual e parte da estrutura de código como hooks, gerenciamento de estados, regras de negócio e organização de componentes há um alto potencial de reaproveitamento do código já implementado no projeto web. Isso contribui

diretamente para a padronização, diminui retrabalho, agiliza o desenvolvimento e facilita futuras manutenções.

A aplicação em React também possibilita uma boa integração dos gestos de pinça e deslizar requeridos pelo projeto, com implementação mais simples, o que economiza tempo e padroniza o sistema para futuras manutenções.

Deste modo, consideramos o ReactJS e o React Native as melhores alternativas em desenvolvimento front-end, dadas sua escalabilidade, manutenibilidade, agilidade na criação de interfaces responsivas e os recursos adequados para o desenvolvimento de aplicações web via navegador e aplicativos móveis, respectivamente.

8.2 8.2 Atualização da Seção Framework (TP3)

No TP3 não foram realizados novos apontamentos pelo professor em relação à escolha dos frameworks. Dessa forma, manteve-se a justificativa apresentada no TP2, considerando que a argumentação técnica já contemplava de forma satisfatória os requisitos funcionais e não funcionais do projeto.

GITHUB da Aplicação:

<https://github.com/Lucas-1234567890/tp3-grupo02>

