

Relatório de Progresso - Desafio QA

Plataforma de Gestão de Atividades

A organização das atividades foi realizada utilizando o **Trello**, onde foram criados quadros e cards para controle do backlog, acompanhamento do progresso e registro de bugs e melhorias identificadas durante os testes.

Link do Trello: [quadro-kanban](#)

Organização das Demandas

As atividades foram organizadas em etapas claras, com foco em manter visibilidade e rastreabilidade do trabalho executado. O fluxo adotado contemplou:

- Levantamento e entendimento das funcionalidades principais do sistema;
- Criação de cenários de teste com foco em fluxos críticos;
- Execução de testes manuais e exploratórios;
- Registro de bugs e sugestões de melhoria no Trello;
- Início da automação dos fluxos prioritários;

Essa organização permitiu acompanhar o avanço das atividades de forma estruturada, mesmo com o prazo reduzido do desafio.

Critérios de Priorização

A priorização das tarefas foi baseada nos seguintes critérios:

1. **Funcionalidades core do sistema** (favoritar, busca de partidas, calendário e melhores momentos)
2. **Impacto direto na experiência do usuário**
3. **Risco funcional**, priorizando fluxos que poderiam comprometer o uso da aplicação

4. Viabilidade técnica dentro do tempo disponível

Com isso, foi possível garantir cobertura adequada dos principais fluxos, mesmo com limitações de tempo.

Dificuldades Enfrentadas

Durante a execução do desafio, algumas dificuldades relevantes foram identificadas:

- **Tempo reduzido**, exigindo tomadas de decisão rápidas e foco no que geraria maior valor para o projeto;
- **Instabilidade do ambiente**, com ocorrência de erros JavaScript não tratados pela própria aplicação (ex.: exceções lançadas durante o carregamento da página), o que impactou diretamente a execução e a repetibilidade dos testes automatizados;
- **Falhas originadas no código da aplicação**, que provocavam a interrupção dos testes ainda na fase de inicialização (beforeEach), impossibilitando a validação completa de alguns cenários sem ajustes adicionais;
- **Seletores inconsistentes ou pouco estáveis**, baseados em estrutura de DOM, ordem de elementos ou textos dinâmicos, o que aumentou a fragilidade dos testes e o risco de falsos negativos;
- **Desafios na automação de testes**, principalmente relacionados a comportamento assíncrono, carregamento dinâmico de dados via API e renderização tardia de componentes, exigindo cuidados extras com sincronização e estratégia de espera.
- **Risco de flakiness nos testes automatizados**, causado por re-renderizações frequentes da interface e mudanças de estado não determinísticas, o que reforçou a necessidade de priorizar robustez em detrimento de volume de testes.

Diante desse contexto, optei por não aumentar artificialmente a quantidade de testes automatizados, priorizando a qualidade da análise, a robustez dos cenários implementados e a identificação de problemas reais da aplicação, em vez de entregar testes frágeis e pouco confiáveis.

Como lidei com os problemas

Para mitigar os impactos dessas dificuldades, adotei as seguintes estratégias:

- **Reforcei a documentação dos cenários de teste**, descrevendo claramente fluxos, pré-condições e comportamentos esperados, garantindo entendimento mesmo quando a automação não era viável no momento;
- **Registrei bugs e oportunidades de melhoria de forma clara e rastreável no Trello**, facilitando a comunicação e a priorização técnica;
- **Ajustei a estratégia de automação**, focando apenas nos **fluxos mais críticos e estáveis**, evitando cenários altamente dependentes de comportamento instável da aplicação;
- **Implementei contornos técnicos conscientes**, como o tratamento controlado de exceções originadas na aplicação, exclusivamente para permitir a execução dos testes e viabilizar a validação de funcionalidades, sem mascarar problemas reais;
- **Tomei decisões intencionais sobre onde investir tempo**, sempre buscando gerar maior valor ao projeto, equilibrando automação, análise exploratória e documentação.
- **Adequação da estratégia de automação**, focando em qualidade, legibilidade e manutenibilidade dos testes, ao invés de quantidade;
- **Avaliação contínua de custo-benefício**, direcionando o tempo disponível para atividades que gerassem maior valor ao projeto e melhor refletissem a realidade do ambiente testado.

Reflexão Final - O que faria diferente

Com mais tempo ou em um contexto real de projeto, eu:

- Investiria na **estabilização dos seletores** em conjunto com o time de desenvolvimento;
- Ampliaria a **cobertura de testes automatizados**, incluindo cenários negativos e regressivos;

- Executaria testes de compatibilidade com mais navegadores e dispositivos;
- Evoluiria a automação para um pipeline de CI/CD;

Mesmo com as limitações, busquei manter uma abordagem organizada, crítica e responsável, priorizando entregas que refletissem qualidade, clareza e valor para o produto.