


Instruções para Pull Request e Merge

Status Atual

- ✓ Código implementado e testado
- ✓ Commits realizados (2 commits)
- ✓ Push para GitHub concluído
- ✓ Documentação completa gerada

 **Aguardando:** Criação do Pull Request e Merge

Link do Pull Request

URL: <https://github.com/orlandobrunet-sketch/base-verification/pull/new/feature/high-priority-improvements>

Passo a Passo para Criar o PR

1. Acessar o Link do PR

- Clique no link acima ou acesse manualmente no GitHub
- Faça login com suas credenciais

2. Preencher Informações do PR

Título Sugerido:

✨ feat: Implementar funcionalidades de prioridade alta (Stats, Temas, Explicações, Conquistas)

Descrição Sugerida:

🎯 Objetivo

Implementar 4 funcionalidades de prioridade alta para melhorar a experiência de aprendizado no NefroQuest.

✨ Funcionalidades Implementadas

📊 1. Sistema de Estatísticas e Desempenho

- Tracking automático de respostas por tema
- Dashboard visual com gráficos de progresso
- Tempo médio de resposta
- Top 5 questões mais erradas
- Histórico das últimas 100 questões

📖 2. Modo de Estudo por Tema

- Modo "Todas as Questões"
- Modo "Revisão de Erros"
- Filtro por tema específico
- Estatísticas por tema

💡 3. Sistema de Explicações Expandidas

- Modal detalhado após cada resposta
- Explicação da resposta correta
- Explicação de cada alternativa incorreta
- Referências para estudo

🏆 4. Sistema de Conquistas

- 10 conquistas únicas de nefrologia
- Notificações visuais animadas
- Página de progresso
- Verificação automática

🤖 Melhorias de UI/UX

- 5 novos botões (3 na tela inicial, 2 durante o jogo)
- 4 animações CSS
- Design 100% responsivo

📈 Métricas

- ****Linhas adicionadas:**** ~720
- ****Funções criadas:**** 15
- ****Modais criados:**** 4
- ****Animações CSS:**** 4

📝 Documentação

- ☒ Relatório completo em ``RELATORIO_MELHORIAS_PRIORIDADE_ALTA.md``
- ☒ Commits descritivos
- ☒ Código comentado

✅ Checklist

- ☒ Código implementado
- ☒ Funcionalidades testadas
- ☒ Design responsivo
- ☒ Compatibilidade com sistema existente
- ☒ Documentação completa
- ☐ Review de código
- ☐ Testes em staging
- ☐ Merge aprovado

🔗 Referências

- Branch: ``feature/high-priority-improvements``
- Base: ``main``
- Commits: 2

3. Revisar Alterações

- Verifique os arquivos modificados na aba “Files changed”
- Confirme que todas as alterações estão corretas

4. Criar o Pull Request

- Clique em “Create pull request”
- Aguarde a criação



Passo a Passo para Merge

Opção A: Merge Direto (Se você tem permissões)

1. Revisar o PR

- Leia as alterações
- Verifique se não há conflitos

2. Fazer o Merge

- Clique em “Merge pull request”
- Escolha o tipo de merge:
 - **Merge commit** (recomendado) - Mantém histórico completo
 - **Squash and merge** - Combina commits em um só
 - **Rebase and merge** - Reaplica commits linearmente

3. Confirmar o Merge

- Clique em “Confirm merge”
- Aguarde a conclusão

4. Deletar a Branch (Opcional)

- Clique em “Delete branch” após o merge
- Mantém o repositório limpo

Opção B: Solicitar Review (Se precisa de aprovação)

1. Adicionar Reviewers

- No PR, clique em “Reviewers” no lado direito
- Adicione os revisores necessários

2. Aguardar Aprovação

- Os revisores receberão notificação
- Eles podem comentar ou aprovar

3. Fazer o Merge após Aprovação

- Siga os passos da Opção A
-

Testes Recomendados Antes do Merge

Funcionalidade

```
# 1. Abrir o jogo localmente
# 2. Responder algumas questões
# 3. Verificar:
- [ ] Dashboard de estatísticas abre corretamente
- [ ] Modo de estudo filtra questões
- [ ] Modal de explicação aparece após resposta
- [ ] Conquistas são desbloqueadas
- [ ] Dados persistem após reload
```

Responsividade

```
# Testar em:
- [ ] Desktop (Chrome, Firefox, Safari)
- [ ] Mobile (iOS Safari, Android Chrome)
- [ ] Tablet
```

Após o Merge

1. Atualizar Branch Local

```
cd /home/ubuntu/github_repos/base-verification
git checkout main
git pull origin main
```

2. Deletar Branch Local (Opcional)

```
git branch -d feature/high-priority-improvements
```

3. Deploy em Produção

- Se houver pipeline de CI/CD, o deploy será automático
- Caso contrário, fazer deploy manual do branch `main`

4. Verificar em Produção

- Acessar o site em produção
- Testar todas as funcionalidades
- Verificar console do navegador para erros

Monitoramento Pós-Deploy

Métricas para Acompanhar

Engajamento:

- Tempo médio de sessão

- Número de questões por sessão
- Taxa de retorno de usuários

Uso das Funcionalidades:

- % de usuários que acessam estatísticas
- % de usuários que usam modo de estudo
- Conquistas mais desbloqueadas
- Taxa de visualização de explicações

Performance:

- Tempo de carregamento dos modais
- Erros no console
- Taxa de bounce



Troubleshooting

Problema: PR não pode ser criado**Solução:**

- Verificar se está logado no GitHub
- Verificar se tem permissões no repositório
- Tentar criar manualmente: GitHub → Pull Requests → New Pull Request

Problema: Conflitos de merge**Solução:**

```
git checkout feature/high-priority-improvements
git pull origin main
# Resolver conflitos manualmente
git add .
git commit -m "fix: Resolver conflitos de merge"
git push
```

Problema: Funcionalidade não funciona em produção**Solução:**

- Verificar console do navegador
- Verificar se todos os arquivos foram deployados
- Limpar cache do navegador
- Verificar localStorage do navegador



Suporte

Desenvolvedor: Orlando Brunet

Email: orlandobrunet@gmail.com

Repositório: <https://github.com/orlandobrunet-sketch/base-verification>

Checklist Final

- ☐ Pull Request criado
- ☐ Descrição do PR preenchida
- ☐ Alterações revisadas
- ☐ Testes realizados
- ☐ Aprovação recebida (se necessário)
- ☐ Merge realizado
- ☐ Branch deletada (opcional)
- ☐ Deploy em produção
- ☐ Verificação em produção
- ☐ Monitoramento ativo

Boa sorte com o merge! 🚀

Documento gerado em 28 de Fevereiro de 2026