

BookScore

Projeto A3

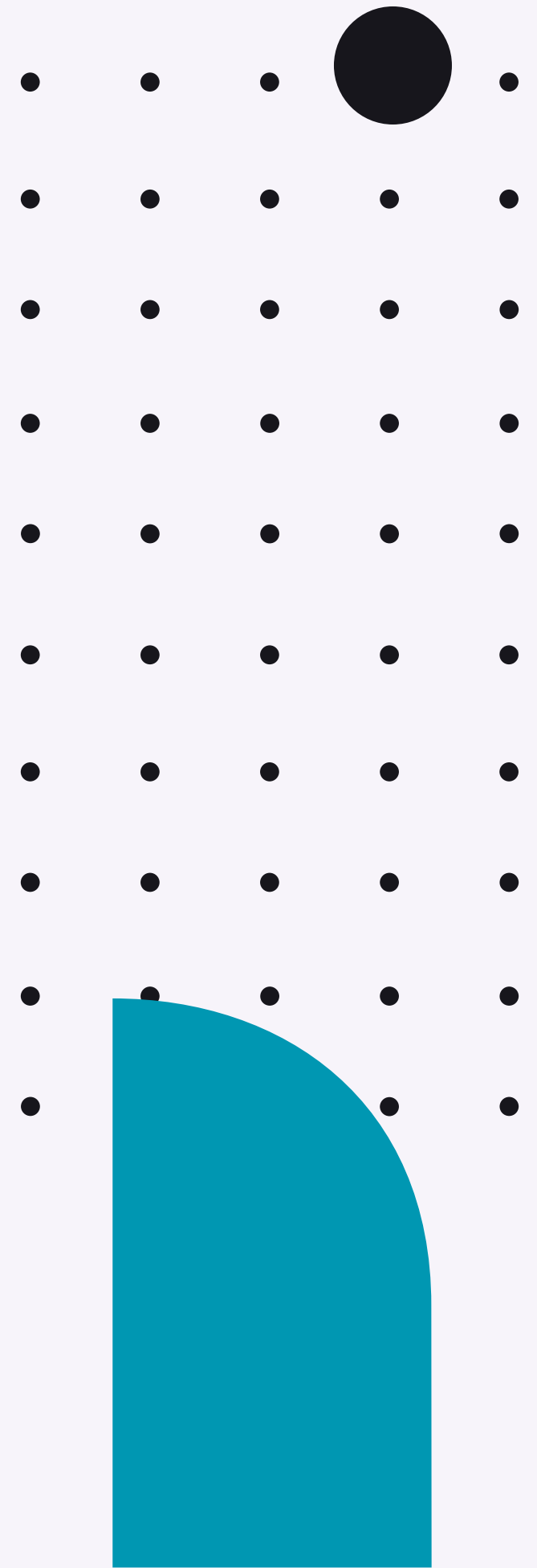
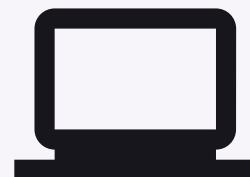
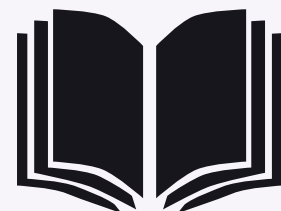
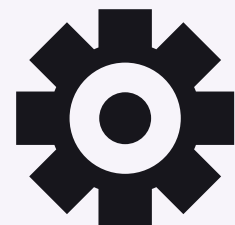
João Vitor Gomes Pereira - 82329432

Mateus Henrique Salvador - 82323463

Felipe Cardoso Silva – 82326693

O que é o BookScore?

- É uma plataforma criada para conectar leitores do mundo todo em torno da experiência de leitura. Nele, qualquer usuário pode cadastrar obras, dar notas de 0 a 10, escrever resenhas e descobrir os títulos mais bem avaliados pela comunidade.
- Desenvolvido em Java, inicialmente como um projeto acadêmico, o BookScore busca evoluir com base em metodologias profissionais de Engenharia de Software, ganhando uma estrutura sólida, amigável e escalável.
- Com um visual intuitivo, performance otimizada e uma proposta clara de valor, o BookScore procura transformar a opinião do leitor em dados inteligentes, oferecendo um ambiente confiável e colaborativo para quem ama livros e quer compartilhar suas descobertas.



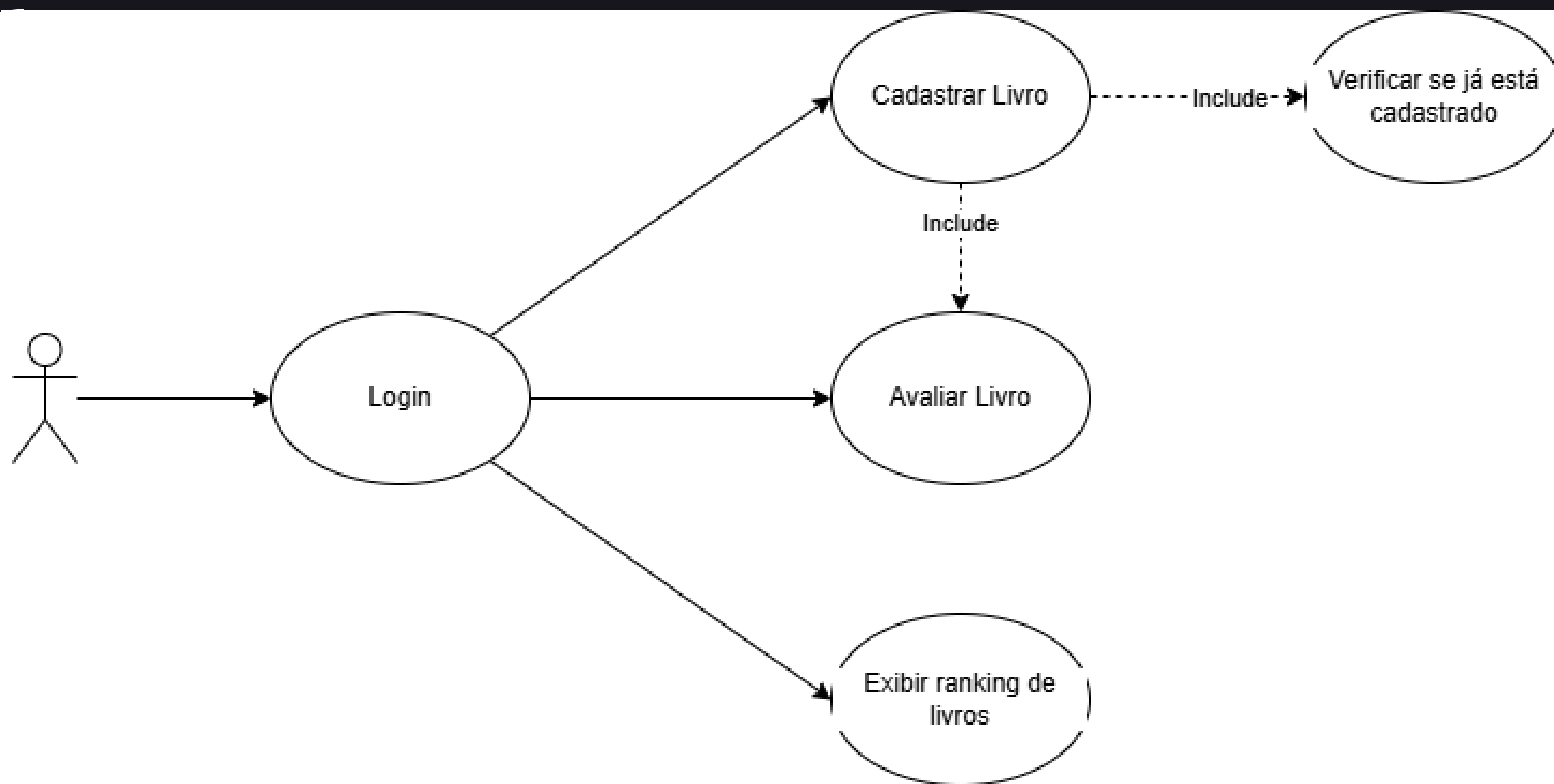
Funcionais

- RF001 – Login: autenticação por e-mail/username e senha, com mensagens de erro claras.
- RF002 – Cadastro: criação de conta com validações e gêneros preferidos.
- RF003 – Logout: opção de sair do sistema com confirmação.
- RF004 – Cadastro de Livros: inserção de livros com checagem de duplicidade.
- RF005 – Avaliação de Livros: nota de 0 a 10 e review opcional, com validação.
- RF006 – Ranking: exibição ordenada por média, número de avaliações e nome.
- RF007 – Gestão de Usuários (Admin): criar, editar e excluir contas com validação.
- RF008 – Feedback: mensagens padrão de sucesso, erro e aviso em todas as ações.

Não Funcionais

- Usabilidade: telas intuitivas, botões com nomes claros e validações visuais.
- Desempenho: operações CRUD em até 2s e ranking em até 3s (até 500 avaliações).
- Confiabilidade: tratamento de erros e transações seguras com rollback.
- Segurança: senha armazenada em texto (projeto acadêmico), com recomendação futura de hash.
- Manutenibilidade: uso do padrão MVC, código comentado e organizado.
- Compatibilidade: compatível com Windows, Linux e macOS com Java 17.

Diagrama de Caso de Uso



Cronograma de Testes

Semana	Fase	Atividade	Duração Estimada	Responsável
1	Planejamento	Definir escopo dos testes, objetivos, critérios de entrada/saída	2 dias	QA / Dev
1	Planejamento	Levantar casos de uso e funcionalidades principais	2 dias	QA / Dev
2	Análise de Testes	Identificar tipos de testes (funcional, usabilidade, segurança etc.)	1 dia	QA
2	Análise de Testes	Elaborar plano de testes (documento)	2 dias	QA
2	Análise de Testes	Definir ambiente de testes (dados, usuários, livros fictícios)	2 dias	QA / DevOps
3	Projeto de Casos de Teste	Criar casos de teste para cada funcionalidade (login, cadastro, etc.)	3 dias	QA
3	Projeto de Casos de Teste	Criar matriz de rastreabilidade (casos x requisitos)	1 dia	QA
4	Implementação dos Testes	Preparar scripts de teste (se automatizado) ou formulários manuais	3 dias	QA / Dev
4	Implementação dos Testes	Revisar e validar os casos de teste com o time	2 dias	QA / Dev
5	Execução dos Testes	Executar testes funcionais (CRUD, navegação, ranking, etc.)	1 semana	QA
6	Execução dos Testes	Executar testes de usabilidade e testes exploratórios	2 dias	QA / Usuários
6	Execução dos Testes	Executar testes de regressão (após correções)	2 dias	QA
7	Registro e Análise de Erros	Documentar bugs encontrados, priorizar e repassar para correção	3 dias	QA / Dev
7	Registro e Análise de Erros	Retestar funcionalidades corrigidas	2 dias	QA
8	Encerramento	Gerar relatório final de testes	1 dia	QA
8	Encerramento	Reunião de lições aprendidas (retrospectiva)	1 dia	Equipe completa



Casos de Testes

Caixa-Preta



TC01-Cadastro de usuário com dados válidos:

Pré-condição:

- App aberto

Passos:

1. Clicar em 'Cadastrar'
2. Preencher campos
3. Enviar

Resultado Esperado:

- Mensagem Sucesso + retorno ao login



TC02-Login com senha incorreta:

Pré-condição:

- Usuário cadastrado

Passos:

1. Preencher login e senha
2. Clicar em entrar

Resultado Esperado:

- Mensagem 'Usuário ou senha incorretos'



Casos de Testes

Caixa-Preta



TC03-Exibição de ranking com dados:

Pré-condição:

- Ao menos 1 avaliação

Passos:

1. Estar logado
2. Clicar em 'Exibir Ranking'

Resultado Esperado:

- Ranking correto + botão Voltar



TC04-Avaliação com nota fora do intervalo:

Pré-condição:

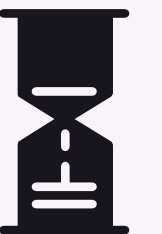
- Livro cadastrado

Passos:

1. Acessar 'Avaliar Livro'
2. Selecionar livro e nota
3. Enviar

Resultado Esperado:

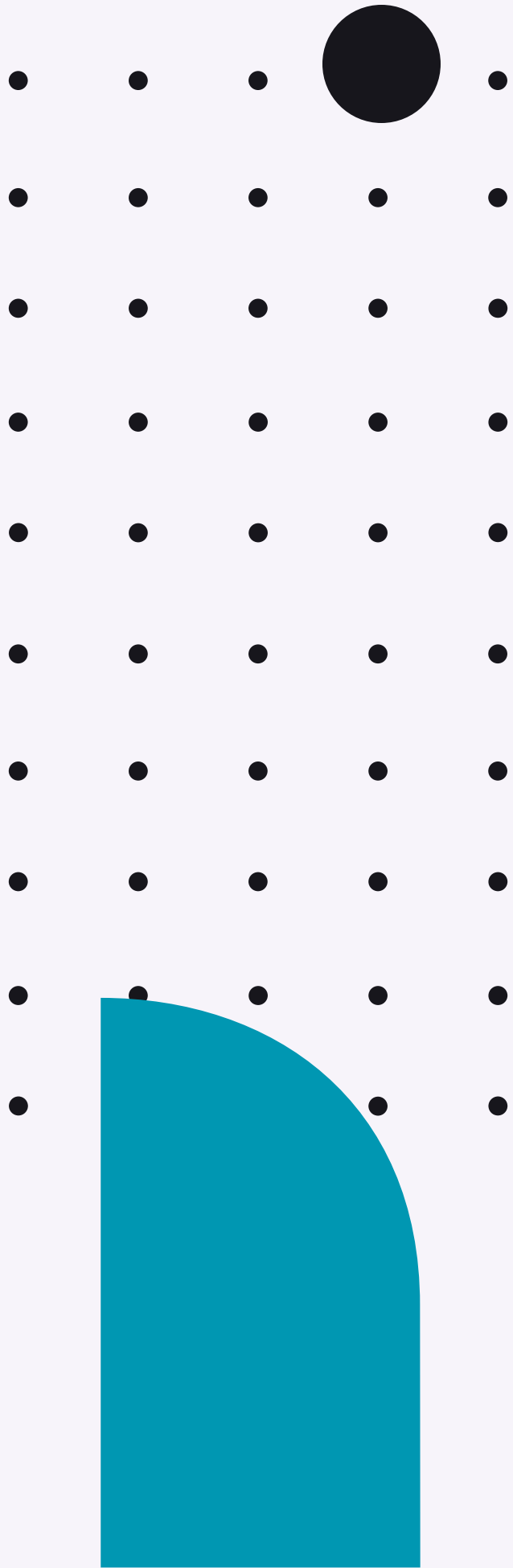
- Mensagem 'Nota inválida. Informe 0-10'



Metodologia Aplicada: Scrum

Para organizar o desenvolvimento e garantir entregas ágeis e iterativas, adotaremos o framework Scrum. Estruturaremos o projeto em sprints curtas, com definição de backlog, papéis bem estabelecidos e entregas incrementais a cada etapa. Durante cada sprint, iremos priorizar funcionalidades essenciais, como cadastro, avaliação e exibição de ranking. A cada ciclo, realizamos revisões e planejamentos para garantir alinhamento com os requisitos e qualidade da entrega final

Sprint	Duração	Atividades
Sprint 1	1 semana	Planejamento do projeto, levantamento de requisitos e definição do escopo.
Sprint 2	1 semana	Modelagem de casos de uso, estruturação do sistema em MVC e prototipagem das telas.
Sprint 3	1 semana	Planejamento de testes, elaboração dos casos e execução dos testes unitários e manuais.
Sprint 4	1 semana	Configuração do repositório, documentação de SCM, encerramento e entrega final.

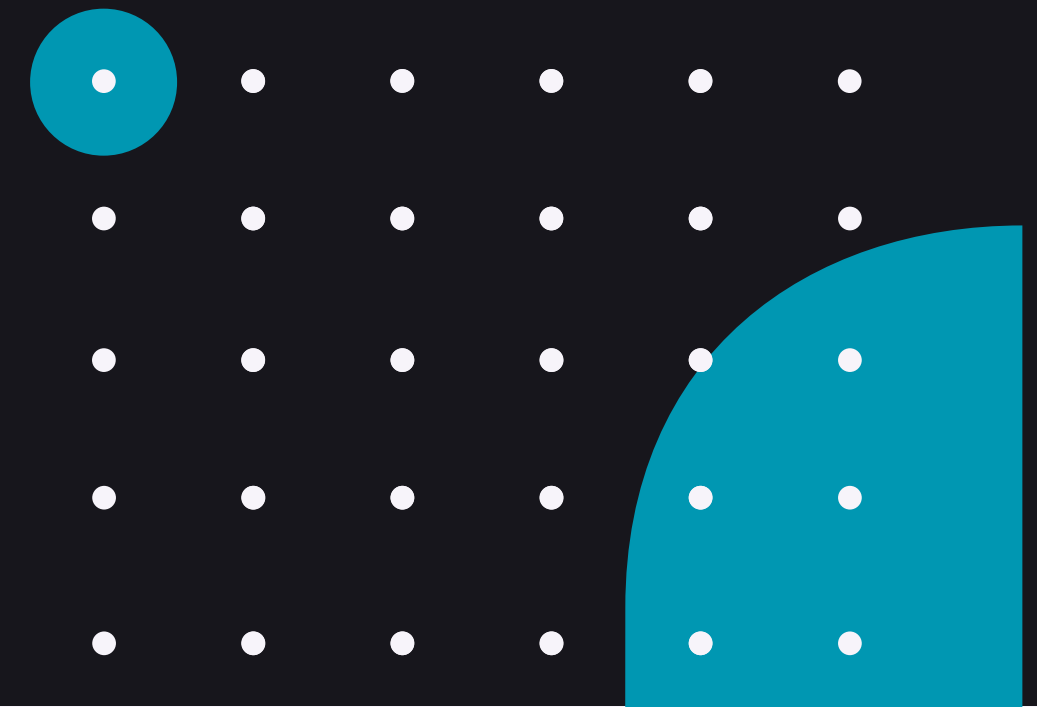


Gestão de Configuração de Software

- Para garantir controle total sobre a evolução do código-fonte e da documentação do BookScore, adotaremos um processo estruturado de Gestão de Configuração de Software (SCM). Todo o projeto será versionado utilizando Git, com repositório hospedado no GitHub, o que permitirá o rastreamento preciso de cada alteração feita ao longo do tempo.
- Cada funcionalidade desenvolvida passará por um pull request, permitindo a revisão de código e a validação de critérios antes da fusão ao código principal.
- Registraremos ECOs (Engineering Change Orders) para alterações mais críticas, documentando o que foi alterado, por quem e por quê. Essa abordagem formal de controle de versão e mudança fortalecerá a rastreabilidade e a manutenção do projeto.



git



Repositório do Projeto

- O repositório do projeto, hospedado no GitHub sob o nome “bookscore”, segue uma organização baseada em arquitetura MVC (Model–View–Controller). Os arquivos estão organizados em três pacotes principais: model (entidades e estrutura de dados), visual (telas e interface gráfica com Swing) e banco (lógica de negócio e operações com o banco de dados).
- Essa separação facilita a manutenção, os testes unitários e futuras expansões do sistema, promovendo um código mais limpo, modular e escalável.

Name	Last commit message	Last commit date
..		
Banco	Entrega06(comum): Visualização de list...	
Model	Entrega06(comum): Visualização de list...	
Resources	Entrega06(comum): Visualização de list...	
Visual	Entrega06(comum): Visualização de list...	
Main.java	Entrega03(gui): Login funcionando	



Benefícios Esperados

Seguindo as técnicas de teste e qualidade durante o desenvolvimento do projeto, esperamos:

1

Redução de falhas em produção

2

Facilidade de manutenção e evolução

3

Rastreabilidade e controle de mudanças

4

Base sólida para escalar o projeto no futuro





FIM



Agradecemos a todos pela atenção e interesse em
conhecer mais sobre o BookScore.

