

Link para os repositórios	
Critérios de Avaliação Coletiva	Avaliação
1 - Práticas Ágeis	
Commits atômicos - o tamanho dos commits estão bons (nem commits incompletos nem grandes demais) e o texto do commit está auto explicativo - dado observado de forma amostral (analisar de 5 a 10 commits)	5%
Documentação das sprints - Todas as sprints estão documentadas	
Documentação das sprints - O grupo lida com as retrospectivas da sprint (os problemas registrados na retrospectiva não se repetem por mais de 2 sprints)	
Documentação das sprints - tem uma análise das métricas pelo scrum master e algum tipo de encaminhamento documentado	10%
Quantidade Total de issues fechadas	
Quantidade média de issues/sprint	10%
Quantidade de Pull Requests fechados	
Quantidade de Pull Requests fechados pro sprint	5%
Mais de 60% das issues possuem comentários >	10%
Qualidade das issues - as issues são bem documentadas, com tarefas e critérios de aceitação, com par alocado. É possível rastrear as decisões do time somente lendo as issues - faça o print de pelo menos uma issue característica do time e coloque no final da planilha	10%
Qualidade dos pull requests - os pull requests são bem documentados, com checklist que guie a revisão do mantenedor.	
Qualidade dos pull requests - os pull requests finais de implementação de funcionalidades possuem testes associados	
Qualidade dos pull requests - os pull requests tem uma boa revisão, com feedback técnico dos mantenedores	10%
Velocity - o velocity do time é estável (tem uma produtividade sem variações abruptas)	
Print do Velocity do time (colocar o print do velocity do time no final da planilha)	
A curva de commits do grupo é ágil - não tem variação/queda abrupta de commits ao longo do tempo no time	5%
2 - Práticas DevOps	10%
Escopo (Roadmap, Story Map, Protótipo)	10%
Escolhas Tecnológicas, Arquitetura escolhida	10%
Deploy	5%
<b>Nota final</b>	<b>100%</b>
Critérios de Avaliação Individual	Informações
Usuário Git	
Quantidade total de commits	30%
Quantidade de issues alocadas	20%
Quantidade de issues alocadas/sprint	10%
Tipo de issue (técnica, documentação, gestão, devops)	15%
Algum feedback negativo pelo time sobre o desenvolvedor (nas retrospectivas, issues) - colocar evidência	

Participação nos rituais (colocar comentario se o participante faltou muito um ritual especifico)	10%
Dedicação semanai (olhar no perfil do git e ver como o participante comita ao longo da semana). O ideal eh commitar pelo menos 3 x na semana	10%
Comunicação - o participante comunica nas issues/PR bem> (olhar de forma amostral)	5%
<b>Nota final</b>	<b>100%</b>
<b>Release 1</b>	
Story Map	10%
Épicos, features, US (critérios de aceitação) - tanto no Story map quanto nas issues (rastreadabilidade). Issues bem documentadas	10%
Protótipos (baixa e alta fidelidade testados antes de implementar)	10%
Código - Implementação de todas as camadas/módulos arquiteturais	30%
Arquitetura Definida/Documentada	10%
Configuração de ambiente - documentação como subir ambiente	10%
Gitpage - visão de produto (com o foco no usuário e não no desenvolvedor)	10%
Padroes Software Livre	10%
Release Notes	5%
<b>Nota Final</b>	<b>105%</b>
<b>Release 2</b>	
Código Fonte - Features, implantação, cobertura de testes, testes de integração	60%
Práticas Ágeis - pareamento, planning, review, retrospectiva, commits atomicos, integração contínua,	30%
Gitpage - visão de produto (com o foco no usuário e não no desenvolvedor) + Release Notes	10%
<b>Nota Final</b>	<b>100%</b>