

**Universidade Federal de Pernambuco :: Centro de Informática**  
**Sistemas de Informação :: Engenharia de Software**  
**Prof. Vinicius Cardoso Garcia**

**INSTRUÇÕES:** Leia as questões com atenção e cuidado, e responda com atenção ao limite de caracteres.

- Esta avaliação tem 3 questões para um total de 10 pontos com **sua resolução sendo individual e sem consulta**.
- Organize o tempo, a prova tem duração de até 2 horas.
- Não é permitido **nada em cima da mesa ou no colo**. Guardem os celulares e demais dispositivos inteligentes, digitais, analógicos e mecânicos no bolso ou na mochila (ou equivalente) e a mesma deve estar no chão.
- Responda todas as questões no **formulário de respostas, de lápis ou caneta**.
- Dúvidas podem ser expostas, **publicamente**, durante os **primeiros 30 minutos** da realização do Exercício Escolar. **Entender o enunciado** faz parte da avaliação.
- Não é permitido ir ao sanitário durante a realização do exercício, **vá antes**.



Você integra a equipe da **Glink Inc.**, uma startup em rápido crescimento que desenvolve o aplicativo **Glink**, voltado à conexão de pessoas do universo geek. Após o sucesso do MVP, a equipe inicia um ciclo de melhorias e expansão de funcionalidades. O time adota Scrum, utiliza Git Flow como estratégia de versionamento e mantém uma arquitetura de microsserviços para permitir evolução independente dos módulos.

O Product Owner (PO) solicitou que o time priorize, para o próximo sprint, três histórias relacionadas à funcionalidade de “Índice de Compatibilidade Geek”, que está gerando alto volume de feedbacks por lentidão na tela de descoberta e inconsistências em alguns cálculos.

**Questão 01 [2,0] – Gestão de Configuração e Controle de Versão**

Durante o desenvolvimento de uma correção para o cálculo do “Índice de Compatibilidade Geek”, dois desenvolvedores da equipe fizeram commits diferentes na mesma classe do serviço de match, ambos atuando sobre o mesmo método. Ao tentar realizar o merge, surgiram conflitos. Explique o que é um conflito de merge e descreva duas boas práticas que poderiam ter evitado ou minimizado esse problema, considerando o uso do Git Flow adotado pela equipe.

**Questão 02 [3,0] – Engenharia de Requisitos e Modelos de Processo**

O PO percebeu que a **user story** abaixo, criada por um estagiário, tem gerado diferentes interpretações entre os desenvolvedores e QA:

Como usuário, quero ver meu índice de compatibilidade com outros perfis para entender melhor quem combina comigo.

**Pergunta:**

1. Identifique **duas lacunas** nesta user story que podem causar ambiguidades.
2. Reformule a história aplicando boas práticas de escrita de **user stories** (seguindo o formato *Como [persona], quero [funcionalidade], para [benefício]*) e **critérios de aceitação claros**.
3. Justifique como a aplicação dessas práticas contribui para a **qualidade do software** e redução de retrabalho.

**Questão 03 [5,0] – Modelos de Processo, Qualidade de Software e Requisitos Não Funcionais**

Após o último *sprint review*, o time detectou que o cálculo do “Índice de Compatibilidade Geek” está impactando negativamente a experiência do usuário: o tempo de resposta na tela de descoberta subiu de **250ms** para **600ms**, ultrapassando o limite não funcional definido.

O Scrum Master propôs uma **retrospectiva técnica**, destacando a necessidade de revisar práticas de qualidade e fluxo de desenvolvimento. O PO, por sua vez, quer entender se o problema decorre de uma **falha de processo**, **má especificação de requisitos** ou **decisões arquiteturais**.

**Pergunta:**

Analise a situação e responda:

1. Quais fatores do **modelo de processo adotado (Scrum + Git Flow)** podem ter contribuído para o problema?
2. Que **ações de melhoria de processo e qualidade** (técnicas e gerenciais) poderiam ser implementadas nas próximas iterações para prevenir problemas semelhantes?
3. Se você fosse o engenheiro de software responsável, como garantiria que **requisitos não funcionais de desempenho** fossem continuamente verificados e atendidos?

Boa sorte!

## Cenário de Avaliação: Estudo de Caso da Startup "Glink"

### Visão Geral do Projeto

Em um mercado digital saturado por aplicativos de relacionamento genéricos, um grupo de empreendedores e entusiastas da cultura pop identificou uma lacuna significativa: a dificuldade de criar conexões verdadeiramente autênticas baseadas em paixões específicas. Para muitos, hobbies como games, animes, RPG de mesa e universos de ficção científica não são apenas um passatempo, mas uma parte central de sua identidade.

Assim nasce o **Glink**, um aplicativo de relacionamento e comunidade projetado desde o início para o público nerd/geek. A plataforma busca ser um ecossistema inovador que conecta pessoas com base em seus interesses, facilitando a descoberta de novos amigos, parceiros para jogar online ou um relacionamento amoroso com alguém que entenda a diferença entre um "Mecha" e um "Kaiju".

O aplicativo visa atender a uma crescente demanda por espaços digitais seguros e segmentados, onde os usuários podem expressar suas paixões sem receio, garantindo uma experiência de descoberta divertida, relevante e focada na compatibilidade de interesses.

### Proposta de Valor e Diferenciais Competitivos (Funcionalidades do MVP)

Para se destacar, o **Glink** focará em quatro pilares centrais:

- Perfil Profundo e Personalizado:** Diferente dos perfis superficiais, o **Glink** permite que os usuários construam uma identidade digital rica.
  - Requisito Funcional Chave:** Implementação de um sistema de tags de interesse com múltiplas categorias (Games, Animes, Filmes, HQs, RPG, etc.), permitindo ao usuário detalhar seus gostos específicos (ex: "JRPG", "FromSoftware", "Studio Ghibli", "Sci-Fi dos anos 80").
  - User Story Exemplo:** "Como um fã de RPG de mesa, eu quero adicionar a tag 'Dungeons & Dragons 5e' ao meu perfil e destacar que sou 'Mestre de Jogo' para que outros jogadores possam me encontrar facilmente."
- Índice de Compatibilidade Geek:** O coração do sistema. Em vez de apenas geolocalização, o algoritmo prioriza a afinidade.
  - Requisito Funcional Chave:** O sistema deve calcular e exibir um percentual de compatibilidade entre dois perfis, com base na sobreposição e relevância das tags de interesse.
  - Requisito Não-Funcional Associado:** O cálculo e a exibição do Índice de Compatibilidade na tela de descoberta não devem adicionar mais de 300ms de latência ao carregamento do perfil.
- Quebra-Gelos Contextuais:** O aplicativo ajuda a iniciar a conversa, diminuindo a ansiedade do "primeiro oi".
  - Requisito Funcional Chave:** Após um match, a tela de chat deve sugerir automaticamente perguntas ou tópicos de conversa baseados nos interesses em comum de maior destaque.
- Comunidade Segura e Moderada:** A criação de um ambiente acolhedor é uma prioridade máxima.
  - Requisito Não-Funcional Chave:** O sistema deve incluir ferramentas de denúncia de perfis e um processo de moderação claro para garantir a segurança e o bem-estar dos usuários, alinhando-se às melhores práticas de qualidade de software no quesito segurança.

### Cenário da Equipe, Processo e Tecnologia

A startup "**Glink Inc.**" opera em um ambiente de alta incerteza e rápida evolução. A equipe inicial é pequena e multidisciplinar, composta por 8 pessoas: 1 Product Owner (PO), 1 Scrum Master, 3 Desenvolvedores Backend, 2 Desenvolvedores Mobile (React Native) e 1 profissional de DevOps/Infraestrutura.

- Modelo de Processo:** A equipe adotou o framework Scrum com sprints de duas semanas para garantir entregas de valor contínuas e capacidade de adaptação rápida ao feedback dos primeiros usuários.
- Gestão de Configuração:** Todo o código-fonte (backend, frontend e scripts de infraestrutura) é versionado em um repositório Git (no GitHub), utilizando a estratégia de Git Flow para gerenciar o desenvolvimento de novas features, correções e releases.
- Arquitetura:** Aprendendo com os desafios históricos de sistemas monolíticos que se tornaram difíceis de manter, a equipe optou por uma arquitetura baseada em microsserviços desde o início, para permitir a evolução independente dos componentes (ex: serviço de perfis, serviço de match, serviço de chat).

Boa sorte!

**Visão de Futuro e Evolução do Produto (Pós-MVP)**

Para manter o engajamento e expandir o modelo de negócio, o roadmap do Glink inclui as seguintes evoluções:

- **Modo "Party Finder":** Uma funcionalidade dedicada para usuários que não buscam um relacionamento, mas sim encontrar grupos para jogar online (LoL, Valorant, etc.) ou para formar uma mesa de RPG.
- **Eventos da Comunidade:** Integrar o app com eventos do mundo real (como a CCXP) ou eventos online (como campeonatos de e-sports), permitindo que os usuários combinem de se encontrar ou participem de atividades juntos. Isso exigiria a coordenação de múltiplas equipes, um cenário ideal para a aplicação de frameworks de escala como o SAFe.
- **Gamificação Avançada:** Implementar um sistema de conquistas e badges (ex: "Conversador Nível 5", "Mestre em Sci-Fi") para incentivar a interação e o preenchimento completo do perfil.
- **Integração com Plataformas de Conteúdo:** Permitir que os usuários conectem suas contas da Steam, PSN, ou MyAnimeList para importar e exibir seus jogos e animes favoritos automaticamente.

Este aplicativo não apenas resolverá a necessidade de conexão dentro de um nicho, mas também tem o potencial de criar uma comunidade vibrante e engajada, tornando-se o ponto de encontro digital definitivo para a cultura geek.