Itajaí, 01 de Junho de 2017

Universidade do Vale do Itajaí - UNIVALI

Centro de Ciências Tecnológicas da Terra e do Mar

Curso de Bacharelado em Ciência da Computação

Disciplina de Engenharia de Software

Professor: Antônio Carlos Silva

Acadêmicos: Arthur Passos, Andriél Ferreira, Eliel Silvério Junior, Fernando Concatto e Hálersson Paris Goes

Projeto: Repositório de Exercícios de Algoritmos de Programação

Requisitos Funcionais

1. O sistema deve permitir que o usuário cadastre uma conta.
2. O sistema deve permitir que o usuário realize *login* em sua conta.
3. O sistema deve permitir que o usuário realize a criação de salas de estudo.
4. O sistema deve permitir que o dono da sala exclua sua sala.
5. O sistema deve permitir que colaboradores da sala convidem usuários para sua sala através de email ou nome de usuário.
6. O sistema deve permitir que membros de uma sala saiam da sala.
7. O sistema deve permitir que administradores da sala atribuam cargos para os membros da sala.
8. O sistema deve permitir que colaboradores da sala removam membros da sala.
9. O sistema deve permitir que colaboradores cadastrem exercícios na sala.
10. O sistema deve permitir que colaboradores da sala criem novas categorias para os exercícios.
11. O sistema deve permitir que o criador do exercício especifique casos de teste durante a criação do exercício.
12. O sistema deve permitir que o criador do exercício cadastre soluções para o exercício.
13. O sistema deve verificar as soluções cadastradas pelo criador do exercício com base nos casos de teste especificados por ele, informando-o sobre as discrepâncias encontradas.
14. O sistema deve informar ao criador do exercício os erros contidos na solução cadastrada por ele, detalhando o tipo e localização do erro.
15. O sistema deve permitir que o criador do exercício especifique dicas sobre o exercício.
16. O sistema deve permitir que o criador do exercício defina a pontuação base do exercício.
17. O sistema deve manter um registro de quem resolveu cada exercício.
18. O sistema deve permitir que o criador do exercício e os administradores da sala excluam exercícios.
19. O sistema deve permitir que o criador do exercício e os administradores da sala tornem o exercício invisível para usuários básicos.
20. O sistema deve permitir que o criador do exercício e os administradores da sala editem os exercícios.
21. O sistema deve permitir que o usuário entre nas salas às quais ele é membro.
22. O sistema deve apresentar a lista de exercícios disponíveis para resolução na sala.
23. O sistema deve permitir que o usuário veja a lista de membros da sala, indicando aqueles que estão *online*.
24. O sistema deve permitir que o usuário filtre (dinamicamente) a lista de membros da sala.
25. O sistema deve permitir que o usuário filtre a lista de exercícios.
26. O sistema deve permitir que o usuário ordene a lista de exercícios.
27. O sistema deve apresentar todos os detalhes do exercício ao ser selecionado pelo usuário.
28. O sistema deve permitir que o usuário escolha uma linguagem ao iniciar a resolução do exercício.
29. O sistema deve permitir que o usuário altere a linguagem durante o desenvolvimento da solução.
30. O sistema deve permitir que o usuário solicite a visualização de dicas sobre o exercício, sob a pena de uma redução na pontuação do exercício.
31. O sistema deve verificar a integridade léxica, sintática e semântica da solução do usuário e informá-lo sobre o resultado do processo.
32. O sistema deve testar a consistência da solução do usuário em relação aos casos de teste cadastrados pelo criador do exercício, informando-o quais passaram e quais falharam.
33. O sistema deve pontuar o usuário assim que uma solução correta for submetida, apresentando a ele a quantidade de pontos obtidos e seu progresso de nível.
34. O sistema deve oferecer uma dica ao usuário ao realizar uma submissão incorreta, informando-o sobre o custo associado.
35. O sistema deve apresentar ao usuário as soluções de referência cadastradas pelo criador do exercício ao término do mesmo.

Requisitos Não Funcionais

1. O sistema deverá funcionar sob navegadores *web* que suportam tecnologias HTML5.
2. As senhas das contas devem possuir de 8 a 20 caracteres.
3. Os identificadores de conta dos usuários devem possuir no máximo 32 caracteres, contendo apenas letras, números, hífens e *underscores*.
4. Todos os campos obrigatórios devem ser demarcados com um asterisco.
5. Caso o usuário insira uma informação inválida em um campo, o mesmo deve ser realçado em vermelho.
6. As opções disponíveis aos usuários devem corresponder com seu nível de permissão.
7. Técnicas de enfileiramento deverão ser utilizadas em caso de sobrecarga do sistema de gerenciamento de compiladores.
8. Os programas submetidos pelos usuários deverão ser compilados por compiladores padrões de cada linguagem, sendo o sistema responsável por invocá-los no momento da submissão de código-fonte.
9. Programas de usuários deverão ser limitados tanto em tempo de execução quanto em consumo de memória ao serem executados pelo servidor que hospeda o sistema. Também deverão ser executados com o menor nível de privilégio possível.

Regras de Negócio

1. Para realizar a submissão de uma solução, o usuário deve previamente submetê-la a uma verificação sintática e semântica, e deve obter êxito neste processo.
2. Para ser aceita como correta, uma solução deve obter sucesso em todos os casos de teste cadastrados (tanto no cadastro do exercício quanto na submissão pelo estudante).
3. Para acessar qualquer parte do sistema que não seja a interface de *login*, o usuário deve estar autenticado.
4. Ao cadastrar uma conta, o usuário deve obrigatoriamente informar seu e-mail, nome, identificador de conta, senha e confirmação de senha.
5. Ao criar uma sala, o usuário deve obrigatoriamente informar o nome da sala, com no máximo 255 caracteres.
6. Ao criar uma sala, o usuário poderá informar zero ou mais e-mails e cargos, indicando os membros iniciais da sala.
7. Usuários só poderão alterar o cargo ou remover membros que estiverem em um nível de permissão abaixo do seu.
8. Ao cadastrar um exercício, o colaborador deve obrigatoriamente informar seu título, enunciado, pontuação base, uma ou mais categorias e no mínimo um caso de teste.
9. A dificuldade dos exercícios deverá estar entre 1 e 5 (inclusivo).
10. A dificuldade padrão na criação do exercício será de 1.
11. Um exercício deverá ter de zero a dez dicas.
12. Um exercício deverá recompensar de 1 a 999 pontos.
13. Um exercício deverá ter no mínimo uma solução de referência.
14. Um exercício deverá ter no mínimo um caso de teste.
15. Um exercício poderá ser editado, excluído ou tornado invisível somente se nenhum usuário tentou resolvê-lo.
16. O criador do exercício deverá especificar a penalidade em pontos para cada dica do exercício.
17. Um usuário poderá resolver um exercício mesmo se já tiver resolvido o mesmo anteriormente, porém não receberá nenhum ponto adicional.
18. O usuário, ao desenvolver a solução, deverá escolher de uma lista predeterminada de linguagens de programação, contendo no mínimo as linguagens Java, C, C++, C#, JavaScript e Python.
19. Os códigos-fonte submetidos por usuários não deverão exceder 1MB de tamanho.
20. Usuários deverão ser associados a um nível, que varia de 1 a 99.
21. O nível de um usuário deverá ser aumentado quando a quantidade de pontos adquiridos por ele exceder o limite do nível atual.
22. Ao atingir o nível 99, o usuário não receberá níveis adicionais, mas a quantidade de pontos em seu perfil deverá continuar sendo incrementada normalmente.
23. As salas de estudo terão um limite superior de 65536 membros.