

## Programa IT Academy – Processo Seletivo – Edição #22

### Instruções para Desenvolvimento e Submissão

Desenvolva uma solução para o problema utilizando a linguagem/ambiente que preferir. Mesmo que não consiga concluir, que faça apenas partes da solução ou que tenha uma solução com erros, faça o envio e entregue o que tiver conseguido fazer.

É obrigatório que você envie todos os materiais necessários para que possamos avaliar sua solução, incluindo:

1. O código e todos os arquivos necessários para que sua solução seja testada;
2. Um arquivo com instruções para execução (ex: readme);
3. O Template de Solução (você recebeu este arquivo como anexo no e-mail onde foi comunicada sua classificação) preenchido conforme orientações nos campos contidos no arquivo. O Template de Solução deve ser submetido em formato .pdf. O Template de Solução deve ser renomeado desta maneira: “SeuNomeCompleto\_Template.pdf” (ex: AndréMoréDeOliveira\_Template.pdf);
4. O Termo de Autoria (você recebeu este arquivo como anexo no e-mail onde foi comunicada sua classificação) preenchido e assinado. Você poderá assinar digitalmente ou fisicamente. O Termo de Autoria deve ser submetido em formato .pdf. O Termo de Autoria deve ser renomeado desta maneira: “SeuNomeCompleto\_Autoria.pdf” (ex: AndréMoréDeOliveira\_Autoria.pdf);

Empacote todos os arquivos (código/solução, instruções para execução, Template de Solução e Termo de Autoria) em formato.zip e faça o envio através do link citado. O arquivo .zip deverá ter tamanho inferior à 50MB. Não inclua no .zip os pacotes de terceiros que foram importados no projeto, nem nada que estiver listado no .gitignore. O arquivo .zip deve ser renomeado da seguinte maneira: “SeuNomeCompleto.zip” (ex: AndréMoréDeOliveira.zip).

A submissão deverá ser feita única e exclusivamente através do link a seguir: [https://pucrs.qualtrics.com/jfe/form/SV\\_6G6tTsa25VXF7Zs](https://pucrs.qualtrics.com/jfe/form/SV_6G6tTsa25VXF7Zs)

O enunciado do exercício técnico se encontra na próxima página. Você deve ler atentamente todas as instruções para que consiga desenvolver a solução. Desejamos boa sorte para você neste processo seletivo!

## Enunciado do Exercício Técnico

### Corrida de Startups - "STARTUP RUSH"

Neste desafio, você irá desenvolver um sistema que simula o *Startup Rush*, um torneio entre pequenas startups que competem por reconhecimento e destaque no mercado. Inspirado em programas de competição, o torneio ocorre por meio de rodadas eliminatórias avaliadas por um "usuário-jurado", responsável por administrar as batalhas e atribuir as pontuações com base em eventos simulados.

#### Contexto

Startups com ideias inovadoras, apresentando nome, slogan e ano de fundação, disputam entre si em rodadas sucessivas. Cada rodada representa um momento de avaliação, podendo ser influenciada por fatores positivos ou negativos simulados pelo sistema. O torneio acontece até que reste apenas uma vencedora, que será coroada como campeã do *Startup Rush*.

#### Regras do Torneio

- Participam entre 4 e 8 startups, obrigatoriamente em número par.
- Cada startup inicia com 70 pontos.
- As rodadas são compostas por batalhas em pares, sorteados automaticamente pelo sistema.
- As batalhas são administradas individualmente pelo usuário (quem conduz o torneio), que escolhe uma batalha ainda pendente e insere os eventos ocorridos.

#### Eventos Possíveis em uma Batalha

Durante a batalha, o usuário pode registrar eventos específicos que afetam diretamente a pontuação das startups:

- **Pitch convincente:** +6 pontos
- **Produto com bugs:** -4 pontos
- **Boa tração de usuários:** +3 pontos
- **Investidor irritado:** -6 pontos
- **Fake news no pitch:** -8 pontos

Esses eventos devem ser atribuídos manualmente pelo administrador da batalha, conforme o desempenho simulado de cada startup. Uma *startup* pode receber um determinado evento uma única vez por rodada, entretanto, um evento pode ocorrer com as duas *start ups* que estão competindo (Por exemplo: A *start up* A e a *start up* B tiveram boa tração de usuários, mas só poderão ter os +3 pontos que representam o evento somados uma única vez em sua pontuação na rodada). Ao fim da disputa, o sistema calcula o vencedor com base na pontuação acumulada. A *start up* vencedora ganhará automaticamente 30 pontos e avançará para a próxima etapa.

## Empates e “Shark Fight”

Se houver empate ao final da batalha, o sistema automaticamente realiza uma *Shark Fight*, uma rodada relâmpago onde uma *startup* recebe aleatoriamente +2 pontos. O novo placar decide a vencedora da disputa.

## Funcionalidades

### [1] Cadastro de startups

- a) Nome, slogan e ano de fundação.
- b) Validação para garantir entre 4 e 8 participantes, obrigatoriamente em número par.

### [2] Sorteio de batalhas e estrutura do torneio

- c) Ao iniciar o torneio, o sistema realiza o pareamento aleatório entre as startups para formar as batalhas da rodada.

### [3] Administração de batalhas individualmente

- d) O usuário escolhe qual batalha deseja administrar.
- e) Exibe os nomes das startups e a pontuação atual.
- f) Permite registrar eventos que impactam a pontuação.
- g) Calcula e apresenta o vencedor.
- h) Realiza o *Shark Fight*, se necessário.
- i) Soma automaticamente 30 pontos na pontuação de cada *start up* vencedora.

### [4] Avanço automático de fase

- j) Quando todas as batalhas de uma rodada forem concluídas, o sistema deve avançar automaticamente para a próxima fase do torneio, sorteando novamente os pares entre as startups classificadas.

### [5] Relatórios e resultados

- k) Exibição de uma tabela geral com todas as startups ordenadas por pontuação final (decrecente), contendo estatísticas individuais: número de pitches, bugs, trações, investidores irritados e penalidades.
- l) Apresentação do slogan da startup campeã ao final do torneio.

**[6] FEATURE EXTRA.** Por fim, você deverá incluir uma funcionalidade extra completamente à sua escolha. Descreva claramente a funcionalidade, faça a sua implementação e a sua demonstração. Use a sua criatividade!

## Observações finais

- a) Você pode exibir as informações solicitadas da maneira que achar mais conveniente e útil, utilizando caracteres, símbolos, números, espaços, interface gráfica, páginas web, etc. Use a criatividade e mostre o que você sabe!
- b) Sugere-se o desenvolvimento de um programa na linguagem de sua preferência, com uma interface também de sua preferência podendo ser gráfica ou textual/console, com um menu com as opções enumeradas nos requisitos; uma interface web bem construída, contudo, contará pontos extra.
- c) Você deve escrever o código que realiza as funções requeridas e armazena os dados lidos em memória (do jeito que você quiser). Não é necessário gravar dados em nenhum formato, nem usar sistemas de banco de dados. No entanto, se você quiser armazenar dados em um banco de dados, isso contará pontos extra.
- d) O programa deverá lidar com dados de entrada inválidos e informar uma mensagem adequada caso ocorram. Lembre-se de demonstrar isso nas capturas de tela ao realizar os testes.
- e) Não esqueça de enviar os resultados desse desafio utilizando o modelo fornecido. Lembre-se também de incluir o link de acesso ao vídeo de apresentação (com até 3 minutos de duração), no qual você deve explicar sua solução.

Bom trabalho.