

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

Nome Completo: Júlia de Moura Schivinski

E-mail: jschivinski@gmail.com

### Exercício Técnico

#### APRESENTAÇÃO EM VÍDEO DA SOLUÇÃO

Grave um vídeo de até 3 minutos apresentando sua solução, sugere-se a divisão o vídeo da seguinte forma: 1,5 minuto demonstrando o software em funcionamento; 1,5 minuto explicando um trecho do código que você considera relevante. Você não precisa aparecer no vídeo, apenas compartilhar a tela e narrar com sua voz.

<https://youtu.be/bVIpM699MAc>

#### RESUMO DA SOLUÇÃO


Dado o desafio proposto foi criado uma solução web usando HTML, CSS e JavaScript para o desenvolvimento.

## APRESENTE OS TESTES DAS FUNCIONALIDADES DESENVOLVIDAS

Insira neste espaço capturas de tela do programa em execução, mostrando todas as funcionalidades desenvolvidas. Explique cada funcionalidade apresentada. Não deve ser incluída imagens do código-fonte. Use o espaço que julgar necessário para tornar sua explicação clara.

### FUNCIONALIDADE 1 – Cadastro de startups

a) Nome, slogan e ano de fundação.



**STARTUP RUSH**

Startup

ex: Estagiários da Dell

Slogan

ex: Tentar nunca desistir

Ano de Fundação

ex: 2005

Número de Participantes

ex: 6

**Cadastrar**

*Coragem para começar,  
ousadia para vencer.*

Foi criado uma página de cadastramento das startups, contendo o nome da startup, slogan, ano de fundação e foi **adicionado** que deve ser informado o número de participantes. Em cada input foi apresentado um exemplo de como e o que deve ser colocado.

- b) Validação para garantir entre 4 e 8 participantes, obrigatoriamente em número par.

**127.0.0.1:5500 diz**

Não é possível salvar pois:

- O número de participantes deve ser entre 4 a 8 e par.

OK

# STARTUP RUSH

Startup

CloudHive

Slogan

Potencial ilimitado. Tudc

Ano de Fundação

2024

Número de Participantes

5

Cadastrar

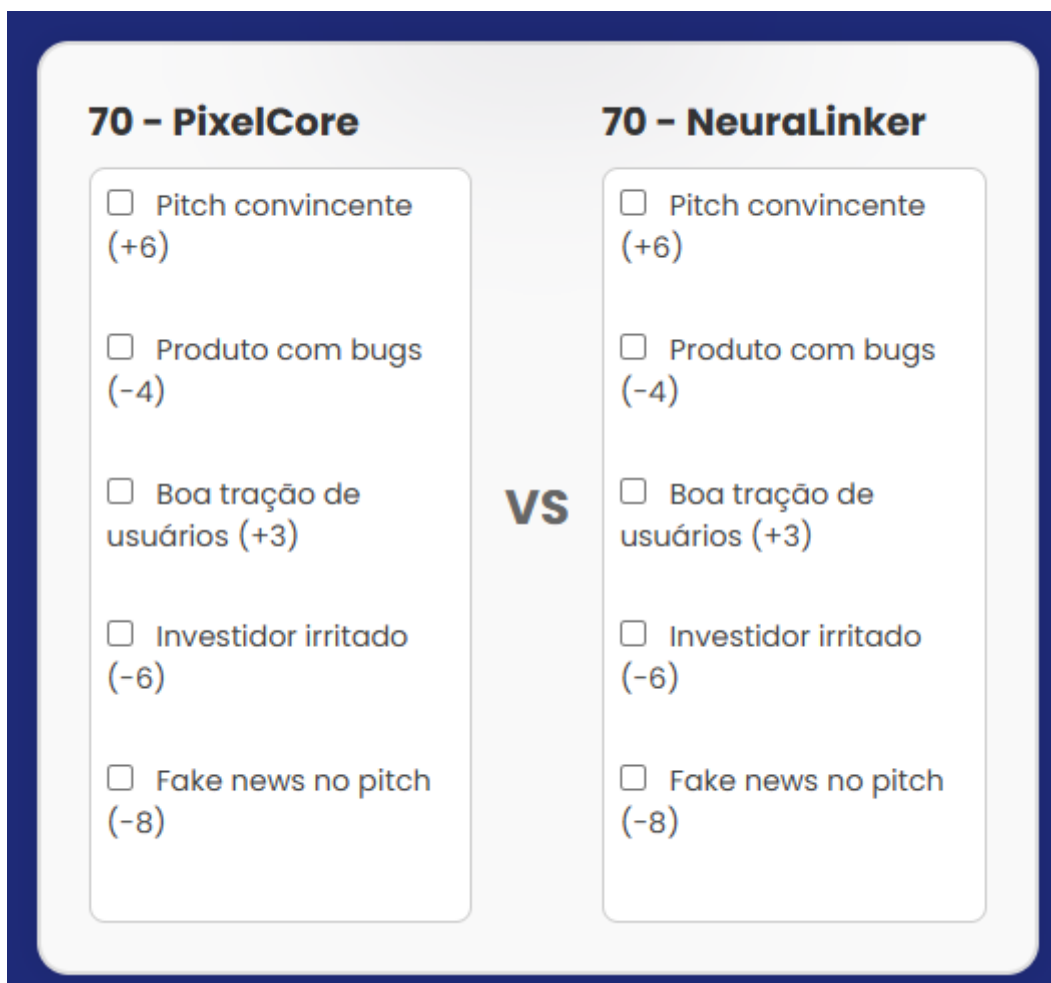
*Coragem para começar,  
ousadia para vencer.*

## FUNCIONALIDADE 2 – Sorteio de batalhas e estrutura do torneio



Como não foi dito exatamente como deveria ser feito optei por seguir pela a solução da seguinte maneira, o sorteio será gerado apenas quando o usuário-jurado clicar no botão responsável por gerar o pareamento aleatório e criar as batalhas.

## FUNCIONALIDADE 3 – Administração de batalhas individualmente



Para exibir as batalhas na tela, decidi posicionar uma por linha, uma abaixo da outra. Essa escolha foi feita para que o usuário-jurado possa focar em uma batalha por vez, evitando distrações com múltiplas informações ao mesmo tempo. Outra coisa que foi percebida durante o desenvolvimento do design, foi que a forma como os cards posicionados um ao lado do outros poderia dar a impressão de que eles estavam conectados de alguma forma, o que poderia confundir o usuário-jurado.

76pts - Nexio	VS	66pts - CodeRise
<input checked="" type="checkbox"/> Pitch convincente (+6)		<input type="checkbox"/> Pitch convincente (+6)
<input type="checkbox"/> Produto com bugs (-4)		<input checked="" type="checkbox"/> Produto com bugs (-4)
<input type="checkbox"/> Boa tração de usuários (+3)		<input type="checkbox"/> Boa tração de usuários (+3)
<input type="checkbox"/> Investidor irritado (-6)		<input type="checkbox"/> Investidor irritado (-6)
<input type="checkbox"/> Fake news no pitch (-8)		<input type="checkbox"/> Fake news no pitch (-8)

Para permitir a adição de eventos, optei pelas checkbox por ser visualmente mais intuitivo para o usuário-jurado. Dessa forma, ele pode selecionar facilmente os eventos correspondentes a cada startup de forma que os pontos são somados automaticamente conforme as opções são marcadas. E caso o usuário-jurado opte por não administrar a batalha manualmente será realizado o Shark figth. Isso por que, sem seleção de eventos, as startups estariam empatadas em relação aos pontos, então nessa situação o sistema irá definir aleatoriamente qual startup avançará para a próxima fase.

FUNCIONALIDADE 4 – Avanço automático de fase

### 70 – PixelCore

- ☐ Pitch convincente (+6)
- ☐ Produto com bugs (-4)
- ☐ Boa tração de usuários (+3)
- ☐ Investidor irritado (-6)
- ☐ Fake news no pitch (-8)

VS

### 70 – Nexio

- ☐ Pitch convincente (+6)
- ☐ Produto com bugs (-4)
- ☐ Boa tração de usuários (+3)
- ☐ Investidor irritado (-6)
- ☐ Fake news no pitch (-8)

Avançar confrontos

Em relação ao avanço automático achei melhor o avanço de fase ser através de um botão, por que se caso o usuário-jurado decidir adicionar mais algum eventos ou retirar algum evento de alguma startup ele possa fazer isso, visto que se fosse automático assim que ele colocasse qualquer opção poderia avançar para a próxima etapa podendo prejudicar alguma das startups.

## FUNCIONALIDADE 5 – Relatórios e resultados

Campeão do Torneio					
Nexio					
Conectando Ideias ao futuro.					
Startup	Pitch Convincente	Produto com bugs	Boa atração de usuários	Investidor irritado	fake news no pitch
PixelCore	2	0	0	1	0
NeuraLinker	0	1	0	0	0
CodeRise	0	1	0	0	0
Nexio	1	0	1	0	0

Ao ser finalizado a última fase o usuário-jurado será direcionado para a tela de resultados e relatórios contendo o nome do campeão, seu slogan e uma tabela com as estatísticas do torneio.

## FUNCIONALIDADE 6 – FEATURE EXTRA

[Home](#) [Administração](#) [Resultados](#) [Nova Startup](#)

Sobre O Startup Rush é um torneio animado onde pequenas startups batalham pelo título de grande destaque no mercado! Inspirado nos programas de competição, a disputa acontece em rodadas eliminatórias, com um "usuário-jurado" no comando. Ele é o responsável por gerenciar as batalhas e atribuir notas conforme os eventos que rolam durante o torneio.

**Regras do torneio**

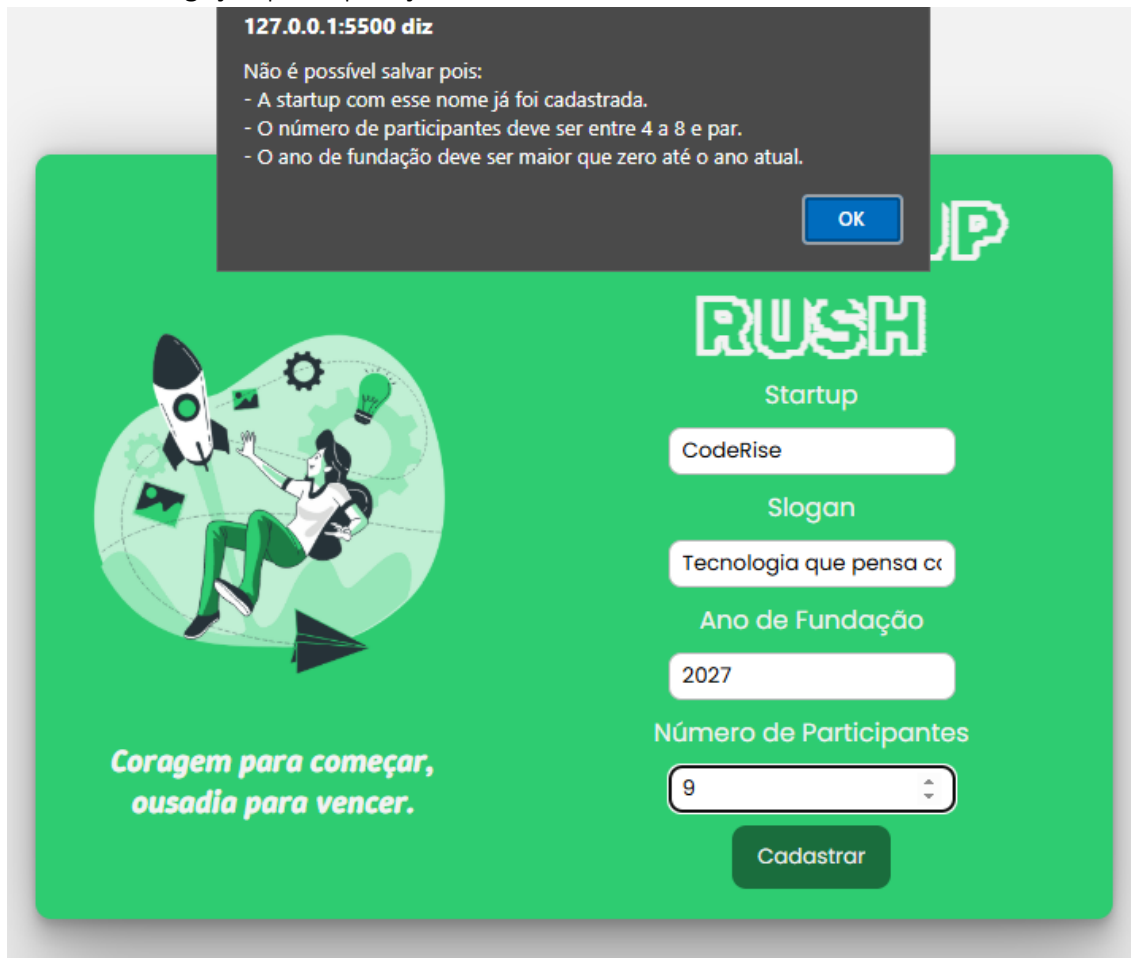
- Participam entre 4 e 8 startups, obrigatoriamente em número par.
- Cada startup inicia com 70 pontos.
- As rodadas são compostas por batalhas em pares, sorteados automaticamente pelo sistema.
- As batalhas são administradas individualmente pelo usuário (quem conduz o torneio), que escolhe uma batalha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

**Possíveis eventos**

- 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

Como optei por desenvolver o projeto em formato web, achei interessante incluir uma página inicial com os tópicos: Sobre o Startup Rush, Regras do Torneio e Possíveis Eventos. Essa estrutura proporciona uma melhor contextualização para o usuário e

facilita a navegação pela aplicação.



The image shows a web application interface for 'RUSH Startup'. On the left, there is a green circular graphic with a rocket, gears, and a person. Below it, the text reads: **Coragem para começar, ousadia para vencer.**

On the right, there is a registration form with the following fields:

- Startup:** CodeRise
- Slogan:** Tecnologia que pensa c
- Ano de Fundação:** 2027
- Número de Participantes:** 9

At the bottom right is a green button labeled 'Cadastrar'.

An error message box is overlaid on the top left of the form, displaying the IP address '127.0.0.1:5500 diz' and the following text:

Não é possível salvar pois:

- A startup com esse nome já foi cadastrada.
- O número de participantes deve ser entre 4 a 8 e par.
- O ano de fundação deve ser maior que zero até o ano atual.

An 'OK' button is located at the bottom right of the error message box.

Nessa funcionalidade foi criada 2 outras validações além da dos números de participantes. Durante o desenvolvimento do desafio percebi a necessidade de criar uma validação a mais para o nome da startup, por que gerava conflitos na hora de somar os pontos das startups, pois ela ficava somando em uma só, apesar de ter duas com o mesmo nome, e validação do ano foi criada pois não é possível que o ano de fundação de uma startup seja anos maiores que 2025.





Durante o desenvolvimento dessa parte final do desafio, percebi que a aba resultados ficava vazia até o final do torneio, por isso pensei em adicionar algumas informações na página inicial, contendo um feedback que não tem informações sobre o campeão e relatório por que o torneio não acabou ou não começou.

## AUTOAVALIAÇÃO

Você concluiu a implementação de 100% das funcionalidades solicitadas?

☒ Sim      ☐ Não

Para as 6 principais funcionalidades solicitadas, como você avalia a sua solução?

Marque um 'X'.

	Inexistente/ Insuficiente	Pouco satisfeito(a)	Satisfeito(a)	Muito satisfeito(a)
Funcionalidade 1				<b>X</b>
Funcionalidade 2			<b>X</b>	
Funcionalidade 3		<b>X</b>		
Funcionalidade 4			<b>X</b>	
Funcionalidade 5			<b>X</b>	
Funcionalidade 6			<b>X</b>	

## Principais dificuldades

Por ter sido a primeira vez que tenho participado de um processo seletivo no qual tenho que desenvolver enfrentei algumas dificuldades iniciais, principalmente em relação a como começar, por isso a solução que encontrei foi descrever cada funcionalidade como eu poderia desenvolver ela explicando de forma detalhada como iria resolver cada uma. Isso me ajudou a visualizar melhor os passos necessários para implementar cada parte do sistema.

Para armazenar os dados, minha primeira ideia foi utilizar o **Firestore**, do **Firebase**. No entanto, encontrei obstáculos, como instalar algumas dependências e como utilizar ele, foi então que percebi isso estava consumindo mais tempo do que eu poderia dedicar, e decidi, estrategicamente, utilizar o **LocalStorage**, que apesar de não ser um banco de dados, ele atendeu bem às necessidades do projeto dentro do escopo proposto.

A funcionalidade que foi mais complexa de resolver foi a *Administração de batalhas individualmente*, por ter várias subfuncionalidades acabei me perdendo e que dificultou a integração da subfuncionalidade de somar os 30 pontos para cada startup. Além disso, fiquei em dúvida se os pontos dos eventos das rodadas anteriores deveriam ser somados ou zerados. Diante da situação, da complexidade e do tempo curto, decidi priorizar outras subfuncionalidades que considerei mais relevantes para o desafio.

## Auto avaliação sobre o desempenho Geral

A entrega deste desafio representou um marco importante na minha jornada como desenvolvedora, por isso atribuo a nota 9,0 ao meu desempenho. Desde o início, busquei me organizar da melhor forma possível para garantir que todas as funcionalidades fossem entregues com qualidade e dentro do prazo. Para isso, utilizei duas estratégias que considero fundamentais: a divisão das funcionalidades em pequenas tarefas — categorizadas como prioritárias ou extras — e a criação de um cronograma diário. Essa estrutura me permitiu distribuir bem as tarefas de lógica, desenvolvidas em JavaScript, e as de design, feitas com HTML e CSS puro, sem o uso de frameworks, equilibrando os dois lados do projeto de forma justa.

Adotei uma metodologia híbrida de Kanban com “sprints” pessoais inspiradas no Scrum. Antes de começar a programar, fazia pequenas “reuniões” comigo mesma, analisando o que havia sido feito, marcando o que já estava concluído no cronograma e definindo as próximas prioridades. Essas revisões diárias me ajudaram a manter o foco, evitar distrações e não adicionar mais tarefas do que o necessário, especialmente em relação ao design, onde era fácil querer ir além do planejado.

Embora o processo não tenha sido fácil, desistir nunca foi uma opção. Enfrentei cada desafio com dedicação e persistência, sempre fazendo o meu melhor dentro do que conheço até agora. No final, entreguei um projeto do qual me orgulho muito. Esse desafio foi mais do que uma experiência técnica — foi também uma experiência de crescimento profissional. Pude

entender como funciono sob pressão, quais métodos de organização mais me ajudam e o que posso melhorar nos próximos projetos.

**Mas por que uma nota 9,0?** Porque sei que alcancei um ótimo resultado, mas também reconheço que ainda há muito a aprimorar — e é justamente isso que me impulsiona a seguir aprendendo e evoluindo.

Por fim, o maior aprendizado que levo é que cada desafio superado é um passo a mais na direção do meu objetivo: conquistar um estágio na área e continuar evoluindo como desenvolvedora. Muito obrigada por essa oportunidade!

Obrigado por participar deste processo seletivo.