

Desenvolver um sistema web capaz de gerenciar os jogos internos do Instituto Federal de Sergipe.

O sistema deve permitir que as inscrições das **equipes** e que possam ser criadas por **esporte** e por **curso**, de forma que apenas uma equipe por esporte/curso.

Cada esporte deve possuir o número máximo e mínimo de cada **atleta**. Por exemplo, futebol de salão, o número mínimo é de 5 atletas e o máximo de 10.

Não há limite de cursos escritos, porém deve haver construção de **evento** voltado para cursos do **integrado, técnico e superior** e que não precisam ocorrer no mesmo período. Desta forma, não haverá confrontos entre integrado, superior e técnico.

O Cadastro da equipe deve estar vinculado ao curso e **Campus**, e permitir a definição do nome da equipe.

Os **jogos** devem ocorrer em uma **data e hora** especificada no sistema e ao final do jogo, o **árbitro** deve alimentar o **placar** com resultado.

Apenas o **atleta “técnico”** poderá cadastrar a equipe do seu time, e este “técnico” deverá ser definido pelo **coordenador** do curso.

Todos os **atletas** devem se cadastrar no sistema previamente utilizando como base o **Nome completo, Apelido, matrícula, número de telefone e senha** para posterior acesso.

Os **coordenadores** já estarão cadastrados previamente diretamente na base e terá como login a **matrícula e uma senha auto-gerada** pelo sistema, que deverá ser “**enviada por e-mail**”. **O envio do e-mail será simulado.**

O sistema deve montar os **grupos** baseado na quantidade de equipes inscritas, de forma que deverá ser múltiplos de 3 equipes por grupo, **de preferência**, e 4 ou 5 (caso haja necessidade). Como mostra o exemplo abaixo:

Nr Equipes	3	4	5	6	7	8	9	10	11	12
Grupo 01	3	4	5	3	3	4	3	3	3	3
Grupo 02				3	4	4	3	3	4	3
Grupo 03							3	4	4	3
Grupo 04										3
Grupo 05										

Os grupos devem ser gerados de forma aleatória pelo sistema, então ao clicar na opção gerar grupos, após a inscrição de todas as equipes. Então caso haja necessidade de refazer o sorteio, bem provavelmente que os confrontos serão refeitos. Exemplo:

Levando em consideração que a Equipe A, B e C caíram no mesmo Grupo após o Sorteio, Equipe A irá jogar Contra Equipe B e a Equipe C fica aguardando o fim do jogo para poder jogar contra a equipe A.

A x B (Confronto = Jogo)

A x C

B x C

- Logo, um grupo com 3 equipes devem gerar sempre 3 jogos.
- Um Grupo com 4 equipes devem gerar sempre 6 jogos.
- Um grupo com 5 equipes devem gerar sempre 10 jogos.

Os jogos podem acontecer seguidos, o importante é seguir a ordem que foi definida e a data e horário planejado de cada jogo.

Após o final dos jogos da fase de grupos (**fase classificatória**) de uma determinada competição de futsal do integrado (ou seja, todos os jogos da fase de grupo foram realizados e placares foram informados no sistema). O organizador do evento poderá gerar a **fase eliminatória** sendo que passam os dois melhores de cada grupo para a fase eliminatória. Sendo assim, usando como sempre uma competição com 4 grupos, serão classificados 2 de cada grupo, gerando um total de 8 equipes portanto, deve ser criada uma fase de quartas de final até a final. (Chave Cheia).

Quartas → Semi → Final → Campeão

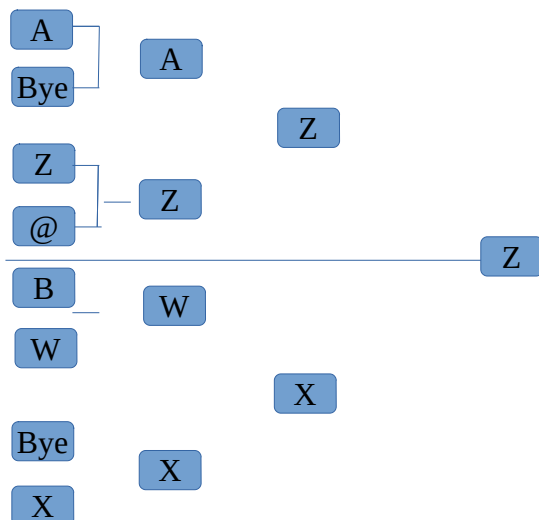
Exemplo de Chave não cheia: 9 equipes, sendo que cada grupo conterão 3 equipes. Passam 2 equipes para a fase eliminatória, sendo um total de 6 equipes.

Nesse caso utilizaremos a fase de quartas para compor as 6 equipes.

Grupo 1: A, B, C → **A, B**

Grupo 2: @, W, Y → **@, W**

Grupo 3: X, Z, K → **X, Z**



O sistema deverá validar se todos os grupos já foram alimentados com os placares ou se houve WO, caso contrário não será permitido gerar a fase eliminatória e aparecerá uma mensagem para o usuário.

O WO poderá ser informado pelo árbitro a uma equipe como também poderá desfazer o WO.

O Sistema deverá classificar os dois mais bem colocados de cada grupo para jogá-los na fase eliminatória.

O melhor colocado do grupo, não poderá enfrentar o seu segundo colocado no grupo a não ser que os dois cheguem na final.

### Observações:

O sistema será dividido em duas partes, back-end e front-end, a primeira unidade corresponde ao back-end apenas, já a segunda unidade, corresponderá ao back-end mais o front-end.

Todo o back-end deverá ser construído em Spring respeitando as camadas básicas: Controller, Service e Repositório.

O front-end deve ser construído desacoplado do back-end de forma que a integração será feito por requesições JSON com o Rest.