

## Trabalho 2 de JavaScript

1- Fazer um APP para registrar e acompanhar um torneio do tipo “todos contra todos”. Limitar a no máximo 20 participantes devido ao grande número de jogos.

### Dados de cadastro

- Perguntar quantos jogadores/times participarão do torneio. Alternativamente pode-se ir perguntando os nomes até que todos estejam inscritos. (Inicialmente só precisa pegar o nome, mas se quiser ou precisar pode pegar mais informações de cada jogador/time).
- Captar e armazenar o nome de todos jogadores.

### Gerar Partidas

- Desenvolver um algoritmo que gere automaticamente as partidas em que cada jogador jogue 1 vez contra cada um dos demais jogadores (torneio de todos contra todos em 1 turno, sem se preocupar com local, mando de campo, etc. Também não precisa se preocupar com rodadas ou ordem em que os jogos são apresentados).

### Mostrar partidas e solicitar os resultados de cada partida.

- Depois de gerar as partidas mostrar todas as partidas em cards (ou outras formas que achar necessário). Junto aos nomes de jogadores de cada partida colocar botões e/ou campos textos para que o usuário informe o resultado da partida (pode ser um torneio de qualquer tipo de esporte/disputa... futebol, xadrez, jogo on-line, tênis... e os dados que devem ser recolhidos vão depender das regras de cada tipo de torneio. Podem ser simplesmente escolher o vencedor. Ou então informar um placar em campos texto, pode ou não ter empate? Quantos pontos vale a vitória? E o empate,... etc).
- As partidas devem ficar sempre visíveis, mesmo depois que seu resultado seja preenchido. A qualquer momento deve ser possível ver todas as partidas; tanto as que já terminaram quanto as que ainda não foram jogadas.

### Contabilizar e mostrar os pontos de cada jogador no torneio

- O App deve disponibilizar um botão (ou outro recurso similar) que ao ser clicado irá ler os resultados de todas as partidas e contabilizar os pontos de todos jogadores. Depois o App deverá mostrar os pontos de todos os jogadores em ordem crescente de pontuação destacando o jogador/time vencedor.

### Armazenar e recuperar dados do torneio no localStorage

- Depois de pegar os nomes dos jogadores, deve ser possível armazenar e recuperar os dados do torneio. Deve ser possível, por exemplo: preencher os resultados de algumas partidas, armazenar estes dados no localStorage, desligar o celular e no dia seguinte reabrir o app recuperar os jogos e seguir com o preenchimento dos resultados de jogos restantes.

### Requisitos adicionais

Os campos para pegar os dados do usuário devem ser **inputs** estruturados e ter seus respectivos labels (Não usar alert, confirm ou prompt). Evite possíveis erros do usuário (impedir inserção de dados errados, apagar/desabilitar campos já respondidos e botões já clicados)

**Caprichar no visual do App**, colocar título, bordas, cards, etc.

**O App deve ser responsivo** e rodar no desktop e no celular:

- deve passar pelo validador da w3c sem erros (<https://validator.w3.org/>). Em todas as fases do App.
- não deve apresentar erros pegos pelo console do JavaScript(JS) dos navegadores.
- Usar css, bootstrap ou outro framework css para dar um bom visual ao App.

O App deve ser disponibilizado no github. O link para o repositório e o github pages do trabalho deve ser **informado no SIGAA até 28/08**.

O trabalho deverá ser **apresentado em aula no dia 29/08**.

## Dicas / Sugestões

### Para gerar jogos

Existem vários algoritmos, um dos mais simples é o seguinte:

```
jogadores=[fulano, beltrano, ciclano, benteno]
```

```
for (i=0, i<=jogadores.lenght-2 , i++) {  
    for (j=i+1, j <=jogadores.lenght-1, j++) {  
        imprimirJogo(jogadores[i], jogadores[j]);  
    }  
}
```

Se você quiser saber quantos jogos serão gerados no torneio basta seguir a fórmula:  
 $\text{numeroDeJogos} = (n*(n-1))/2$ , onde  $n$  é o número de participantes.

## Estruturas de dados

Vai precisar de um **array de objetos jogador** com dados de cada jogador (nome e pontuação por exemplo).

Provavelmente este array será percorrido para gerar os jogos e mostrar os resultados do torneio.

Vai precisar de um **array de objetos jogo** com dados de cada jogo (jogadores, resultado, etc).

Provavelmente este array será percorrido para contabilizar a pontuação dos jogadores. Outra alternativa é adicionar os pontos de um jogo na medida que seus resultados são preenchidos.

Estes arrays são candidatos a serem armazenados e recuperados no localStorage pois com eles pode-se refazer o andamento de um torneio.

## Armazenamento no localStorage

Foi disponibilizado no sigaa um arquivo com a implementação de uma “Todo List” que armazena e recupera dados de uma todo list no localStorage. Algo semelhante deve ser feito com os dados do torneio (provavelmente com as 2 estruturas de dados citadas acima).

Ao recuperar os dados do localStorage deve-se percorrer o array de jogos e gerar novamente os cards dos jogos armazenados (com e sem resultados).