

DISCIPLINA: JAVASCRIPT BÁSICO
PROF. WILTON DE PAULA FILHO
PROVA DE PROFICIÊNCIA

A prova de proficiência consistirá em:

- **[NOTA: 40,0 pontos]** Desenvolver as aplicações 1 e 2 de forma assíncrona. Os enunciados dessas atividades estão disponíveis neste documento. As soluções deverão ser hospedadas e os endereços públicos de acesso a elas deverão ser enviados para o email do professor Wilton Filho (wiltonpaulafilho@gmail.com) até o dia 14/01/24;
- **[NOTA: 20,0 pontos]** Apresentar as soluções das aplicações 1 e 2 ao professor Wilton Filho. A data da apresentação será agendada pelo professor logo após o envio das soluções.
- **[NOTA: 40,0 pontos]** Desenvolver uma solução para um problema cujo enunciado será disponibilizado no dia 16/01/24 (quinta-feira). A solução deverá ser implementada no laboratório 07 do IFTM campus Uberlândia Centro das 19h às 22h30;

APLICAÇÃO 1/2

Objetivo geral

Construir uma página web para gerar convites de casamento automaticamente, a partir do preenchimento de um conjunto de informações.

Interface + Formulário

A página inicial (index.html) deverá ser composta por um formulário contendo os mesmos elementos e *layout* mostrados na imagem a seguir.

Título do Convite:	<input type="text" value="Convite de casamento"/>
Frase:	<input type="text" value="A sua presença é especial em nosso casamento"/>
Convidados:	<input type="text" value="Wilton, José, Ana"/>
Data:	<input type="text" value="10/09/2012"/>
Endereço:	<input type="text" value="Rua Logradouro Alameda, 100"/>
<input type="button" value="Gerar Convites"/>	

Ao clicar sobre o botão “Gerar Convites” todos os convites (um para cada convidado, conforme os nomes informados no campo “Convidados”) deverão ser exibidos ao usuário numa nova página HTML (UTILIZE ARMAZENAMENTO LOCAL PARA RESOLVER ESSE PROBLEMA). O layout dos convites, inclusive dos textos, deverão ser os mesmos das imagens abaixo. Os convites deverão ser centralizados horizontalmente na janela do navegador e posicionados um abaixo do outro, conforme mostrado nas imagens a seguir.



CADA CONVITE DEVERÁ SER EXIBIDO UM APÓS O OUTRO APÓS UM INTERVALO DE TEMPO DE 2 SEGUNDOS, OU SEJA, SE HOVER 5 CONVIDADOS A PÁGINA GASTARÁ 10 SEGUNDOS PARA EXIBIR TODOS OS CONVITES.

APLICAÇÃO 2/2

Construir um jogo da memória contendo as seguintes características:

1. O jogo deverá apresentar uma tela inicial contendo um breve manual (orientações) sobre a utilização do mesmo, uma caixa de seleção com três níveis de dificuldade e um botão que será utilizado para iniciar a partida.
2. Níveis de dificuldade: Fácil (12 cartas), Médio (20 cartas) e Difícil (36 cartas)
3. O tabuleiro do jogo deverá ser composto pela quantidade de cartas correspondente ao nível de dificuldade, conforme mostrado no layout da figura abaixo (nível médio):



4. A escolha das imagens de cada uma das cartas ficará a seu critério.
5. O jogo deverá ter um cronômetro que será utilizado para marcar o tempo total gasto pelo usuário para finalizar a partida. **OBS:** Quando o jogador finalizar a partida, uma caixa de diálogo deverá aparecer solicitando a ele o seu nome completo. Em seguida, estas informações (nome e tempo gasto para completar a partida) deverão ser gravados no dispositivo do usuário (use LocalStorage). Toda vez que a página do jogo (tabuleiro) for carregada deverão aparecer o

nome e tempo do jogador com a melhor pontuação (isso para cada nível de dificuldade, ou seja, no nível fácil deve aparecer o nome e pontuação do melhor jogador nesse nível e assim sucessivamente).

6. O jogador com a melhor pontuação será aquele que errou menos, ou seja, virou menos cartas (menos tentativas). Além do cronômetro, o jogo deverá informar o número de tentativas realizadas pelo usuário. Uma tentativa equivale a uma par de cartas viradas. **OBS:** Esta informação também deverá ser gravada na seção e apresentada ao usuário sempre que ele iniciar uma nova partida.
7. O jogador poderá reiniciar o jogo a qualquer momento. Quando o jogo for reiniciado as cartas serão embaralhadas novamente.
8. Sempre que uma carta for clicada (virada) pelo usuário um som deverá ser produzido. Caso ele acerte, um som diferente deverá ser emitido. O mesmo deverá ocorrer quando ele errar (som diferente do acerto).