



## Web development 2025/1

Global Solution	2o.SEM/2025	CURSO:	ENGENHARIA DE SOFTWARE
DISCIPLINA:	Web Dev	PROFESSOR:	Marcelo Amorim

### Global Solution FIAP 2025 – O Futuro do Trabalho

#### Web Development - Turmas: 1ESPR e 1ESPS

Bem-vindos a este desafio empolgante que aplicará todo o conhecimento adquirido ao longo do semestre!

### Formulário Interativo - Conectando Tecnologia e o Futuro do Trabalho

Nesta atividade, você irá integrar conhecimentos de desenvolvimento web (HTML e JavaScript) com as ideias de projeto interdisciplinar desenvolvidas sobre o tema **O Futuro do Trabalho**.

O mercado de trabalho está em constante transformação, impulsionado por tecnologias como IA, trabalho remoto e a ascensão de novas habilidades (*upskilling*). O projeto escolhido por você em outras disciplinas (ex: uma plataforma de *reskilling*, um sistema de gerenciamento de *freelancers*, ou um aplicativo de saúde mental para trabalhadores remotos) requerirá, em algum momento, a interação com o usuário.

**Seu desafio é criar uma página web simples (HTML/JavaScript) que contenha um formulário de inscrição, cadastro ou registro de interesse que seja a porta de entrada para a sua ideia de projeto.**

## Objetivo

Desenvolver uma página HTML contendo um formulário que utilize **JavaScript puro** para validação de dados e interação dinâmica com o usuário, demonstrando o domínio dos seguintes conceitos:

1. Criação e manipulação de **Funções**.
2. Uso e gestão de **Vetores (Arrays)**.
3. Técnicas de **Manipulação de Strings**.
4. Acesso e modificação de elementos DOM usando `document.getElementById()`.

NÃO É NECESSÁRIO A PERSISTÊNCIA DOS DADOS, APENAS  
VALIDAÇÃO.

## Requisitos do Formulário

Seu formulário deve ter um título claro relacionado ao seu projeto sobre o Futuro do Trabalho (ex: “Inscrição para o Programa FutureSkills”) e incluir os seguintes campos:

Campo Obrigatório	Tipo de Dado	Função no Contexto do Projeto
<b>Nome Completo (CPF)</b>	Texto	Identificação básica.
<b>(e-mail)</b>	Texto	Requer manipulação de <i>strings</i> para validação.
<b>Lista de Habilidades</b>	Select/Radio	Lista de Habilidades
<b>Tipo de Interesse</b>	Select/Radio	Exemplo: “Trabalho Remoto”, “IA”, “Sustentabilidade”, etc.
<b>Botão de Adicionar Habilidade</b>	Button	Deve adicionar uma habilidade na lista de habilidades.
<b>Lista de Habilidades Adicionadas</b>	Área de Lista*	Listar habilidades. Requer armazenamento em <b>vetor</b> .
<b>Botão de Submissão</b>	Button	Deve chamar a função principal de validação do JS.
<b>Área de Feedback</b>	<div> ou <span>	Onde os resultados da validação e o resumo são exibidos.

\* **Área de Lista:** pode ser uma div, um ul, ou o que achar melhor (sua preferência).

## Requisitos de JavaScript (Mandatório)

O script do seu formulário deve cumprir **todos** os requisitos de implementação abaixo:

### 1. Uso de Funções: (exemplos)

- **validarFormulario():** A função principal, chamada ao clicar no botão de submissão. Ela deve coordenar todas as validações.
- **checkEmail(valor):** Uma função dedicada para validar ou normalizar um campo de e-mail, se contém @, .com ou converter-lo para minúsculo, etc..
- **checkCPF(valor):** Uma função dedicada para validar ou normalizar um campo de CPF.
- **exibirFeedback(vetorErros):** Uma função para receber uma lista (vetor) de erros ou um resumo de sucesso e exibir na **Área de Feedback** (Requisito 4).

### 2. Uso de Vetores (Arrays):

- O sistema deve utilizar um **vetor** para armazenar as **Habilidades** do usuário. O aluno deve ser capaz de adicionar pelo menos 3 habilidades a este vetor através de um botão de “Adicionar Habilidade” antes de submeter o formulário.
- A função **validarFormulario()** deve utilizar um **vetor de erros** para armazenar todas as mensagens de erro de validação encontradas e passá-lo para a função **exibirFeedback**.

### 3. Manipulação de Strings:

- Implementar, dentro de **validarFormulario()**, a verificação de formato para os campos, garantindo, por exemplo, que ele siga um padrão mínimo (ex: tamanho, presença de caracteres específicos). Use funções de checagem descritas anteriormente.

### 4. Manipulação do DOM (`document.getElementById()`):

- Todos os dados devem ser lidos usando `document.getElementById('id_do_campo').value`.
- Todas as mensagens de erro ou de sucesso devem ser escritas no elemento **Área de Feedback** usando `document.getElementById('id_do_feedback').innerHTML = ....`

## Avaliação

Critério	Peso	Descrição
<b>Funcionalidade Básica</b>	40%	O formulário carrega e o botão chama a função JS sem erros no console.
<b>Requisitos JS</b>	40%	Uso correto e demonstrável de Funções (3), Vetores (2 usos), Manipulação de Strings (1 uso).
<b>Estrutura HTML</b>	10%	A página HTML é bem estruturada e os elementos estão acessíveis via id para o JavaScript.
<b>Clareza do Código</b>	10%	Código JS bem indentado e com comentários que explicam a lógica principal.

**Observação:** O foco da avaliação é na **funcionalidade do JavaScript**. O uso de CSS é **opcional**, mas o formulário deve ser minimamente legível e utilizável.

## Entrega

- Repositório Github: Crie o projeto dentro de uma Organização com seu time como colaboradores
- Repositório **DEVE OBRIGATORIAMENTE** ter um arquivo **README.MD** contendo nome e RM dos alunos integrantes, descrevendo detalhes do projeto, instruções de uso, requisitos, dependências e demais informações relevantes ao projeto. Será avaliada a clareza e organização do conteúdo apresentado.
- JavaScript: Utilize apenas JavaScript puro (vanilla) - sem frameworks
- Apenas um membro da equipe deve realizar a entrega por meio da respectiva tarefa criada para esta atividade no Teams.