



Documentação Jogo de Palpites para o suporte técnico

Visão Geral

Este projeto é um jogo interativo baseado em tentativas, cujo objetivo é adivinhar um produto com base em suas características. Ele simula um jogo estilo "Termo" e "Loldle", mas com atributos técnicos de produtos Intelbras. O jogador insere palpites e recebe feedback visual sobre o quão próximos estão os atributos do palpite em relação ao produto alvo.

Estrutura HTML

Cabeçalho (<head>)

- Define o título da página e a responsividade via meta viewport.
- Contém uma folha de estilos embutida com toda a estilização da interface.
- Usa fontes padrão e tema escuro com tons de verde e preto.

Corpo da Página (<body>)

- Contém o título principal (Spot by Specs).
- Possui botões de controle: palpar, novo jogo, acesso ao manual, botão de informações (i) e modo desenvolvedor (⚙️) para inserir códigos de debug.
- Campo de busca com sugestões automáticas.
- Tabela onde são exibidos os palpites e os resultados comparativos.
- Modal de popup usado para exibir mensagens de vitória ou derrota.
- Inclusão de scripts externos (html2canvas, produtos.js) e scripts inline para lógica principal do jogo.

Estilos CSS

Tema Escuro Personalizado

- Fundo escuro com detalhes em verde e cinza claro.
- Botões estilizados com hover, focus e animações.

Tabela de Palpites

- Feedback visual com cores distintas para cada tipo de resultado:
 - exact (verde)
 - partial (amarelo)
 - none (vermelho)
 - up/down (ícones de seta)

Modais e Tooltips

- info-btn: mostra informações do desenvolvedor ao passar o mouse.
- popup: modal exibido ao fim do jogo com opção de exportar resultado.

Efeitos Visuais

- Animações como bounceUp, bounceDown, fadeIn e fall (chuva de emojis) enriquecem a experiência do usuário.

Funcionalidades JavaScript

Inicialização

- Define constantes e variáveis globais como MAX_TRIES, PRODUCTS, target, entre outras.
- Seleciona elementos do DOM e configura eventos.

Botões e Eventos

- guessBtn: avalia o palpite atual.
- newGameBtn: reinicia o jogo com novo produto.
- manualBtn: abre o manual em uma nova aba.
- infoBtn: mostra tooltip com informações de autoria.
- devBtn: exibe input para senha de desenvolvedor (eusoudev) que revela o produto alvo (modo debug).

Sugestões Automáticas

- Enquanto o usuário digita, são mostradas até 3 sugestões baseadas no nome do produto.
- Ao clicar em uma sugestão ou pressionar Enter, o palpite é processado.

Mecânica do Jogo

- **Função reset():**
 - Escolhe novo produto aleatoriamente e limpa a tabela.
- **Função evaluate():**
 - Compara os atributos do palpite com os do produto alvo e classifica o resultado:
 - evalList(): compara arrays (tipo, autenticação, rede, etc.)
 - evalNum(): compara valores numéricos (capacidade de usuários)
 - evalIP(): verifica similaridade do grau de proteção IP
 - evalFreq(): analisa proximidade de frequência (com margem de erro)
 - evalText(): compara nomes de fornecedor por inicial
- **Função render():**
 - Cria uma nova linha na tabela com os resultados estilizados conforme o tipo de acerto.
- **Função evaluateAndRender():**
 - Toca som de palpite.
 - Avalia, renderiza e verifica condição de vitória ou derrota.
 - Se venceu: toca som, mostra modal e emojis comemorativos.
 - Se perdeu: toca som, mostra modal e emojis de fracasso.
- **Função emojiRain():**
 - Cria emojis animados caindo na tela para efeito visual.
- **Funções showPopup() e closePopup():**
 - Controlam o modal de resultado final com campos para exportação.

Componentes Externos

produtos.js: script externo que contém o array PRODUCTS, com os dados dos produtos e seus atributos.

html2canvas: biblioteca usada para exportar resultados como imagem.

Estrutura de Dados dos Produtos produtos.js PRODUCTS

Tipo

Array de objetos JavaScript (Array<Object>)

Finalidade

Armazena informações detalhadas sobre uma lista de produtos. A estrutura visa facilitar a manipulação, filtragem e exibição desses dados em aplicações web, sistemas de gerenciamento ou catálogos técnicos.

Estrutura de Cada Objeto

Cada item do array PRODUCTS representa um produto com as seguintes propriedades:

Campo	Tipo	Descrição
nome	string	Nome comercial do produto.
tipo	string	Tipo do item: geralmente "controlador" ou "software".
capacidade	number string	Capacidade de usuários ou registros que o produto suporta.
auth	Array<string>	Métodos de autenticação suportados (ex: "Tag", "Senha", "Biometria").
com	Array<string>	Tipos de comunicação disponíveis (ex: "RS 485", "Wiegand", "IP").
graulP	string	Grau de proteção IP (Ingress Protection), ou "N/A" se não aplicável.
freq	number Array<number> string	Frequência(s) de operação (ex: 125, 13.56 MHz), ou "N/A".
rede	Array<string>	Tipos de conectividade de rede (ex: "Ethernet", "wi-fi", "LAN", "USB").
fornecedor	string	Marca ou fabricante do produto.

Observações Técnicas

Flexibilidade de Tipos: Alguns campos como freq e capacidade admitem múltiplos tipos (number, string, Array<number>) para abranger diferentes formatos de dados, idealmente padronizável.

Valores "N/A": Vários campos usam o valor "N/A" para indicar Nao se Aplica. Em aplicações práticas, pode ser útil tratar esse valor com lógica condicional para evitar erros de tipo ou comportamento inesperado.

Coesão e Coerência: A estrutura é coesa, com consistência semântica entre os objetos, o que facilita filtragens como:

```
const apenasComBiometria = PRODUCTS.filter(p => p.auth?.includes("Biometria"));
```

Facilidade de Expansão: A estrutura é extensível: novos produtos podem ser adicionados facilmente e novos campos podem ser incluídos, se necessário.

Em caso de duvidas fico a disposição

att:

Igor Rodrigues da Silva

Igor.Rodrigues@intelbras.com.br