*– Robson Júnior –*

**JavaScript Power**

a única linguagem em **100%** das páginas web

**Do problema a solução**

O objetivo deste caso é entender como uma linguagem criada em apenas 10 dias há 21 anos atrás se tornou a linguagem mais popular no mundo da tecnologia web, se tornando a primeira e única estar presente em 100% da nossas páginas web.Aqui, será estudado também como essa linguagem ganhou tanto poder que além de estar em todas páginas web já se encontra em aplicações que não dependem de um browser, como NodeJS, MongoDB. Este estudo também tem como propósito se tornar algo como “Um guia rápido” sobre o que é e o caminho que o JavaScript levou e também seus diversos tipos de soluções para variados problemas.

Um dos maiores problemas no inicio da Internet era como trabalhar com páginas dinâmicas de forma simples, onde pudessemos manipular o DOM, pois as páginas eram apenas HTML com links para outras páginas. Além disso, toda e qualquer requisição ou trabalho de modificação deveria ser feito pelo servidor.

O JavaScript realmente foi um divisor de águas. Com ele, foi aberto um novo mundo de possibilidades desde simples animações até compelxos jogos, também desde uma simples requisição para um servidor até ferramentas arrogantes de robustas para ataques hackers e crackers. De agora em diante sugiro que apertem os cintos porque iremos entrar em um caso de estudo do poderoso JavaScript.

É importante ressaltar que neste estudo não houve debate, ele é formado com o embasamento em cima de pesquisas somados a 3 anos de experiência de larga escala focado em JavaScript.

**JavaScript**

Hoje(ano 2016) fez 21 anos que o JavaScript foi criado. Muito confundido com Java devido ao seus nomes escrito de formas iguais nas quatro primeiras letras foi desenvolvido pelo na época Engenheiro de Software Brendan Eich quando parte da Netscape.

No final de 1995 foi implementado pela primeira vez no Netscape onde teve diversos nomes como Mocha, LiveScript para só aí vir o JavaScript.

Em muito pouco tempo, o JavaScript se espalhou entrando no Internet Explorer aproximadamente 8 meses depois de sua primeira implementação no Netscape. No mesmo ano, a Netscape submete o JavaScript para a Ecma Internacional e o resultado disso em pouco tempo depois foi uma versão já padronizada chamada ECMAScript.

Desde então tem sofrido uma contante e rápida evolução, ela é quem hoje é responsável pelos nossos Ajax, iterações com usuários, dinamismo na experiência do usuário entre outras magias.

**Cliente Side**

Cliente side é quem irá interpretar o script para depois rodar e ter determinado resultado, o mais comum browser. Hoje todos os browsers tem o JavaScript implementado em sí, cada um com sua engine. O Firefox com seu SpiderMonkey, o Chrome e o seu poderosoV8, Chakra dentro do Internet Explorer e assim por diante, cada um responsável por sua implementação.

A não muitos anos atrás, era possível apenas simples operações no browser

como alertas, validações e algumas difíceis manipulações no DOM.

Entrando nesse link irá da pra ter uma noção de tamanha evolução aconteceu: <https://trackingjs.com/> .

**Como posso usar o JavaScript?**

*<script type=”text/javascript”>*

*alert(‘Oi mundo!’);*

*</script>*

Uma das maiores habilidades do JavaScript é poder de adaptação e isso soa muito bem para uma adoção ser feita de forma feliz.

Ok, mas o que isso quer dizer?

Isso diz que o JavaScript pode ser encontrado nos diversos conceitos de programação, como:

- **orientada a protótipos**

ele usa protótipos como mecanismo de herança

- **orientada a objetos**

nossa clássica abstração do mundo real para programação

- **orientada a eventos**

disparados através de gatilhos, sendo possíveis criar métodos que são emissores de eventos sendo disparado somente quando necessário

- **programação funcional**

existe funções aninhadas, de primeira classe também podendo ser explorados conceitos de map, redulce, filter.

Além disso, ,também é de tipagem dinâmica assim como quase todas linguagens scripts.

Aaah, claro… e não podemos esquecer que também podemos usar o JavaScript para melhorar a experiência de um usuário com deficiência visual.

**Hack / XSS**

XSS é Cross-site scripting, é um ataque baseado em injeção de algum código em determinado lugar. Esse ataque é possível quando o o ambiente explorado são aplicações web.

Existem dois tipos de XSS.

**Reflected**

O XSS refletido é aquele em que o usuário consegue manipular a página em tempo de execução, por exemplo alterando uma URL depois de achar um valor sendo passado como parametro que entra no DOM.

**Storage**

É aquele ataque que o código é postado e armazenado no servidor, tipo sistemas de comentários. Caso uma vez o script sendo postado junto com algum comentário e inserido na página, ele será propagado a todos que acessarem a página. O script está armazenado ali, por isso chamado storage.

**- formas de explorar**

entre as mais conhecidas está formulários em paǵinas web com falhas na validação, onde se consegue injetar um script no value de algum elemento.

**- exemplos de aplicações comuns**

twitter é um cenário perfeito, onde podemos submeter um texto, caso um código fosse postado com sucesso, teriamos um XSS storage onde quem acessasse a página sofreria o ataque

**- como se proteger**

uma das formas mais eficazes em programação na hora de validar algum valor ou determinado input é pensar em validação inversa. O que isso quer dizer? Quer dizer que ao invés de você validar o que você não deseja se inverte, você irá validar apenas o que deseja negando assim todo o resto

**NodeJS /** Server Side

NodeJS é JavaScript rodando no lado servidor. Criado em 2009 foi desenvolvido baseado em eventos e um processamento não bloqueante. O NodeJS tem a npm como seu principal gerenciador de pacotes.

O NodeJS foi construido sobre a plataforma do Chrome, isso mesmo, o V8, isso e tudo o que mostrado antes garante um processo leve e rápido.

Aplicações comuns para usar NodeJS são aquelas que possui um grande volume de I/O, como chat, sistemas de comentários entre outros. O NodeJS consegue suportar todo esse volume graças ao seu trabalho assíncrono que consegue realizar, que veio evoluido ainda mais agora com uso de promise e async functions.

Hoje temos também grandes cases de mercado que possuem pentagigas de tráfegos de informação por dia como:

- Chat Terra

- NetFlix

- PayPal

- Wallmart

- Google

**MongoDB**

MongoDB é JavaScript escrito no banco de dados. MongoDB é um banco de dados não relacional, voltado a alta performance com seu modelo de dados sempreno formato JSON. O MongoDB foi lançado também em 2009, no mesmo ano do NodeJS. Por baixo dos panos roda um C++. O MongoDB é free e open source. A forte adoção dele é justamente seu schema ser JSON como já trabalhamos na forma natural em JavaScript.

**Revisão**

Então entendemos que o JavaScript se popularizou devido a sua versatilidade e integração a todos ambientes, desde o browser até o banco de dados.

Vimos um pouco de:

- sua história

- onde aplicamos

- uso malicioso

- armazenamento de dados

- cases de uso

*“Always bet on* ***JS*** *– Brendan Eich*”

**Referencias:**

- [https://braziljs.org/#artigo](https://braziljs.org/" \l "artigo)

- <http://nomadev.com.br/>

- <https://pt.wikipedia.org/wiki/JavaScript>

- [https://www.owasp.org/index.php/Main\_Page](https://pt.wikipedia.org/wiki/JavaScript)

- <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

- <https://speakerdeck.com/robsongajunior/infinity-power>

- <https://pt.wikipedia.org/wiki/MongoDB>

- https://nodejs.org/en/

- <https://www.mongodb.com/>

- https://en.wikipedia.org/wiki/MongoDB