

**Desenvolvimento de Aplicações Orientadas a Objetos**

**Prof. Ms. Vinícius Magnus - @vinimagnus**

**Atividade semipresencial**

O JavaScript segue especificações do EcmaScript (<https://www.ecma-international.org/>), que por sua vez lançou a versão atual na especificação do ES 2022 de que foi lançada em 2021, mas já tem a atualização e padronização para [ES 2023](https://tc39.es/ecma262/) que conta com boas atualizações previstas. Outra também com mudanças significativas foi as especificações em 2018 com com ES9, porém a versão com maior mudança foi a de 2015 com o ES6. Desde então vem constantemente realizando debates sobre aprimoramento do padrão devido ao seu alto uso e a necessidade de se adaptar ao desenvolvimento moderno de aplicações, que são altamente debatidos pelos padrões utilizados em bibliotecas e frameworks.

Pesquise e relate quais foram as principais mudanças de ES em suas últimas versões (2015 até 2022) descrevendo os impactos nos padrões de desenvolvimento.

nome: Guga Padilha

prof: Vini Magnus

Durante muito tempo a linguagem JavaScript não recebeu grandes atualizações, mas em **2015** isso mudou. Com a versão do EcmaScript 6 **(ES6),** a linguagem teve uma revolução tão impactante (e tão extensa) que ainda não foi absorvida completamente pela comunidade.

Com a chegada das **arrow functions,** a agilidade foi maior e mais sucinta mas a diferença não é apenas visual. Enquanto funções normais (ES5) possuem o contexto do seu **this** atrelado a ela mesma, arrow, os poderosos **MAP, FILTER E REDUCE** e a chegada das variáveis **CONST e LET**.

**ECMAScript 2017** — não deve ser confundido com ES7 (também conhecido como **ECMAScript 2016** — traz várias melhorias notáveis ​​para a linguagem. Espera-se que os navegadores implementem esse padrão totalmente no verão de 2017, mas o compilador Babel JavaScript já suporta vários recursos hoje. Se você faz uso extensivo de JavaScript e sua base de código está em desenvolvimento ativo, recomendamos que você adicione o Babel ao seu pipeline de compilação e comece a usar os recursos suportados.

O **ES2018** ainda traz outras novidades e melhorias, como novas funcionalidades para expressões regulares (RegExp) e uma revisão nos template literals.

Novidades no **ECMAScript 2019 (ES2019)/ES10**

O método **flat()** cria um novo vetor recursivamente com todos os elementos do sub-vetor concatenados nele até a profundidade especificada.

O método **flatMap()** primeiramente mapeia cada elemento do vetor usando uma função e mapeamente para, em seguida, nivelar o resultado em um novo array. O flatMap é bastante útil, pois mescla um mapa seguido por um **flat** de profundidade 1 em um método mais eficiente.

O método **trimStart()** remove espaços em branco do início de uma string. O método **trimEnd()** remove espaços em branco do final de uma string.

O método **toString()** retorna uma string representando o código-fonte da função. Espaços em branco, novas linhas e comentários serão removidos quando você fizer isso agora, eles serão retidos com o código-fonte original.

Dentre as principais novidades do **ES2020** estão: Importação dinâmica, onde é possível usar a importação como função em qualquer uma das partes do seu código;*BigInt, que ajuda a representar números maiores do que 2^53-1. Promise.allSettled (), que retorna uma determinada promessa; globalThis, em que não será mais necessário diferenciar se o código está, ou não, rodando em um navegador*

*.*Dentro da lista de **ECMAScript de 2021,** mostrando-se tão essencial quanto o comando SQL, foram adicionados alguns novos recursos:*LogicalAssignmentOperators;String.prototype.replaceAll();Promise.any();WeakRefs;Underscores (\_), utilizando separadores em number literals e em bigint literals.*

Você pode ter uma ideia de quais outros recursos podem entrar no **ECMAScript 2022**, observando quais atingiram o estágio três, onde a especificação foi aprovada e os testes foram aprovados. Mas essas propostas precisam ser testadas em uma implementação para ver como elas realmente funcionam na prática.

Por exemplo, a proposta de poder trabalhar ao contrario do array, começando com o último elemento da maneira que você pode em Python está atualmente implementada como um polyfill, em uma tentativa de descobrir se ele colide com qualquer framework JavaScript comumente usado.