

DEC - Dicionário de Erros Comuns

DEC - Dicionário de Erros Comuns	1
Vulnerabilidade	1
Links com url de um site externo	1
Code Smells	2
Using desnecessários	2
IF com apenas uma linha sem usar parênteses	2
Instância de campos private readonly não utilizadas	2
Variáveis não utilizadas instanciadas ou não	3
Saiba utilizar const	3
Bugs	3
Atente-se a sintaxe do HTML5	3

Vulnerabilidade

Erros de vulnerabilidade são relacionados à segurança da aplicação

Links com url de um site externo

Deve se usar `rel="noopener noreferrer"` na tag

Motivo:

rel: Segundo o W3 Schools, o atributo `rel` especifica a relação entre o documento atual e o documento/recurso vinculado ou “linkado”. O atributo “`rel`” é uma abreviação de “`relationship`”, ou “relação” em português e serve basicamente para informar a relação entre uma URL e uma página, ou para ser mais claro a relação que o seu site tem com os demais sites que são linkados a partir de você.

noreferrer: O uso da `rel noreferrer` também não possui peso algum em SEO (Search Engine Optimization), não de forma direta. Esta relação não informa a origem do link (referral link) dessa forma o link não contabiliza um acesso de referência, que popularmente é visualizado no painel do Google Analytics para contabilizar os acessos de seu site à sites externos.

noopener: A `noopener` também não possui peso em SEO, esta relação está no mesmo grupo que a `rel noreferrer` sendo possível usá-las juntas. Esta relação basicamente inabilita o uso do código JavaScript `window.opener.location = newURL`. Este recurso pode consumir demasiados recursos do seu site. Felizmente, os navegadores mais atualizados não permitem por padrão o uso do `window.opener` porém pode ocorrer em versões antigas ou desatualizadas.

Dessa forma também há um mínimo ganho de performance caso o link seja aberto em um `target="_blank"` pois pode carregar consigo algum tipo de código JavaScript pesado.

Code Smells

Using desnecessários

Embora o uso desnecessário não mude nada para o aplicativo produzido, remova-os:

- Ajudará a legibilidade e manutenção.
- Ajudará a reduzir o número de itens na lista de preenchimento automático do IDE durante a codificação.
- Pode evitar algumas colisões de nomes.
- Pode melhorar o tempo de compilação porque o compilador tem menos espaços de nomes para procurar
- quando resolve tipos.
- A construção falhará se esse espaço para nome for removido do projeto.

Dica: Utilize o comando Ctrl+R+G no Visual Studio para remover.

IF com apenas uma linha sem usar parênteses

Na ausência de chaves, a linha imediatamente após uma condicional é aquela executada condicionalmente. Por convenção e boas práticas, essas linhas são recuadas. Na ausência de chaves e recuos, a intenção do programador original não é totalmente clara e talvez não seja realmente o que é executado.

Além disso, é altamente provável que esse código seja confuso para os mantenedores.

Ao invés de:

```
if (condition)
DoTheThing();
```

Utilize:

```
if (condition)
    DoTheThing();
```

Instância de campos private readonly não utilizadas

Os campos private readonly usados apenas para armazenar valores sem lê-los posteriormente é um caso de armazenamento morto. Portanto, alterar o valor desse campo é inútil e provavelmente indica um erro grave no código.

Variáveis não utilizadas, instanciadas ou não

Um armazenamento morto ocorre quando uma variável local recebe um valor que não é lido por nenhuma instrução subsequente. Calcular ou recuperar um valor apenas para substituí-lo ou jogá-lo fora pode indicar um erro grave no código. Mesmo que não seja um erro, é na melhor das hipóteses um desperdício de recursos. Portanto, todos os valores calculados devem ser utilizados.

Saiba utilizar constantes

Um conceito básico é o de constantes. Um campo estático que não é constante nem somente leitura não é seguro para threads. O acesso correto a esses campos a partir de diferentes segmentos precisa de sincronização com bloqueios. A sincronização inadequada pode levar a resultados inesperados; portanto, os campos estáticos visíveis ao público são mais adequados para armazenar dados não alterados compartilhados por muitos consumidores.

Para impor essa intenção, esses campos devem ser marcados somente leitura ou convertidos em constantes.

Ao invés de:

```
public static double Pi = 3.14;
```

Utilize:

```
public const double Pi = 3.14;
```

Bugs

Atente-se a sintaxe do HTML5

Não utilize a tag <i> para ícones, utilize (pode não funcionar de acordo com o tema);

Não utilize a tag para negrito, utilize ;

Adicione aria-describedby para descrever uma <table>;

Adicione scope="col" em <th> das colunas de uma <table> para descrever seu escopo