Boas Práticas de Programação e Codificação.

As boas práticas em programação existem para que todos os desenvolvedores sejam capazes de ler, identificar e corrigir os códigos de terceiros.

Ou seja, essas boas práticas são, na verdade, medidas que tornam o código universal e claro, seja na nomenclatura ou na forma como são separados, por exemplo.

Por isso nesse texto apresentaremos as mais importantes boas práticas em programação.

·Escolher nomes intuitivos;

·Comentar o código;

·Separar os códigos;

·Fazer a indentação do código;

·Fazer testes;

·Não esquecer do backup .



Escolher nomes intuitivos



A função do nome é justamente deixar aquela classe, variável ou método identificável, ou seja, é preciso escolher um nome que indique de maneira direta e clara do que se trata aquela parte.

Embora pareça um pouco óbvio é comum encontrar códigos com nomenclaturas genéricas como 1, 2 e 3 ou A, B ou C.

Comentar o código



O comentário é muitas vezes necessário para lembrar do que é que se trata aquele código, para deixar claro o seu objetivo.

Muitas vezes a nomenclatura adequada já pode ajudar nessa questão, mas o comentário é essencial para qualquer complementação necessária.

Nesse caso é recomendado que o comentário seja direto e inserido no início de tudo.

Separar os códigos



Um programador organizado tende a ter seus códigos bem separados, porém muitos outros não costumam fazer isso, o que torna a localização dos códigos complicada.

Fazer a indentação do código



Fazer a indentação do código, ou seja, identificar através de espaços aonde começa e termina cada bloco, é tido por muitos profissionais como uma prática essencial e básica na hora de programar.

Fazer testes



Os testes são essenciais para garantir que o projeto está funcionando corretamente e que você não terá que fazer nenhuma correção em um futuro próximo.

Não esquecer do backup



Nada pior que estar no meio de um projeto e perceber que perdeu tudo. Então não deixe de fazer o backup dos seus projetos, de preferência na nuvem, isso garante que nenhum mal tempo ou falha no hardware vai levar seus códigos embora.

**Versionamento**

**O que é versionamento?**

O versionamento é um método adotado pelos programadores com o objetivo de controlar e acompanhar todas as mudanças de um projeto. Esse método permite saber quaisquer alterações feitas no projeto e identificar as mesmas. Além disso esse projeto atinge também os usuários finais, permitindo que estes sejam capazes de reconhecer qual a versão do projeto assim como as mudanças que foram feitas no mesmo.

Durante ou após o termino do projeto, sempre é necessário manter uma atenção constante, devido as constantes mudanças que podem afetar o projeto. É exatamente esse controle que define o versionamento.

**Para que serve o versionamento?**

Se feito de forma adequada o versionamento fornece uma maior otimização para o projeto e melhora a segurança do mesmo. Veja alguns exemplos abaixo:

**Análise de ameaças**

É fato que existe atualmente uma grande quantidade de ameaças externas que surgem todos os dias possibilitando grandes prejuízos para o projeto, e o versão permite uma grande proteção contra o mesmo.

**Reparação de erros**

Novas atualizações estão sujeitas a acarretar em possíveis erros no projeto e pode gerar alguns problemas, porém com a ajuda do versionamento é possível regredir em uma das versões anteriores do projeto e corrigir esse erro.

**Alteração de arquitetura**

Por conta da velocidade na qual a tecnologia evolui, sempre à uma grande demanda na implementação de novas funções ou ferramentas no projeto, mas graças ao versionamento é possível acompanhar essa evolução de maneira mais fácil e prática.

**Estética**

A estética do projeto é algo de extrema importância no projeto, pois existe uma necessidade de que permita uma experiência melhor e mais agradável para o usuário final e o versionamento permite fazer essa mudança apenas através de algumas pequenas alterações na devida versão do projeto.

**Como implementar o versionamento?**

Não existe exatamente um padrão obrigatório para o versionamento, permitindo que o responsável pelo projeto construa o projeto de uma forma que fique melhor para ele, porém é recomendável construir o projeto de uma maneira bem estruturada e que permita uma fácil compreensão permitindo um melhor acompanhamento.

Depois de montar uma base, é necessário criar uma ordem hierárquica para identificar a evolução no histórico do sistema. Esta hierarquia é definida geralmente por números em uma ordem crescente.

**Ferramentas de versionamento:**

Para implementar o versionamento, é preciso ter acesso à ferramentas adequadas para o trabalho, conheçam algumas das ferramentas de versionamento mais usadas:

•CVS;

• Subversion;

• TFS;

• GIT;

•Mercurial.

BREAKPOINTS

Breakpoints (pontos de interrupção), são os pontos de quebra onde o layout será ajustado para atender diferentes resoluções. A partir destes pontos é que será possível estabelecer comandos CSS para que o site se comporte de forma adequada visualmente para cada dispositivo.

Os pontos de interrupção são muito úteis na hora de descobrir um erro em um código. Isso se torna ainda mais útil quando ele (o código) é muito extenso, chega a ser impossível encontrar o erro. Esse “erro”, não é necessariamente algo que faça o código não funcionar, e sim algo que faça ele não funcionar da maneira que ele foi criado/pensado para ser. Um computador lê o código linha por linha, então se você criar um código com 50 linhas e comandos, e errar algo na linha 17 por exemplo, o seu código será lido e executado até a linha 16 normalmente, mas para saber qual comando dos vários que você colocou ali nas 50 linhas estão fazendo o seu código falhar, um breakpoint será útil de uma forma absurda.

Com ele, você irá conseguir separar o seu código em quantas vezes precisar, e poderá fazer diversos testes vendo até aonde ou até qual comando o seu código está funcionando. Dessa maneira, você irá descobrir com muito mais facilidade qual das 50 linhas está fazendo o comando da 17° falhar e conseguirá concertar o seu código fazendo ele ter o funcionamento desejado. Em um código um com cerca de 1000 linhas, um breakpoints pode lhe “salvar” de um jeito absurdo.

Breakpoint :

O código irá parar na linha aonde o ponto de interrupção foi colocado, e você irá conseguir encontrar o erro e dar prosseguimento a ele.