**DESENVOLVEDOR DE SISTEMAS: GUIA PROFISSIONAL**

**Cleber Cavalheiro Duarte**

**Gabriel de Oliveira Becker**

**Leonardo Flores Rodrigues**

**Marcos Simão Schneider**

**Tutor Externo: Flavio Rocha de Avila**

Centro Universitário Leonardo da Vinci – UNIASSELVI

Análise e desenvolvimento de Sistemas (ADS0067) – Prática módulo I

08/12/17

**RESUMO**

*No presente artigo será citado a história do desenvolvedor de software. De como ele se tornou importante para a sociedade, será relatado também as suas funções e divergências no setor, será discutido também a ética e sua importância, tal como a importância na segurança do desenvolvimento de software.*

**Palavras-chave: desenvolvedor, código-fonte, ética na computação**

**1 INTRODUÇÃO**

Neste artigo será descrito um guia para a profissão de desenvolvedor de sistemas, onde será relatado a história do desenvolvedor, suas linguagens de programação, sua presença no mercado, de como o desenvolvedor deve lidar com a ética, sua postura perante a sociedade e seu papel, exemplificando condutas antiéticas e como devemos eliminá-las e protegermos os sistemas de tais vulnerabilidades e de pessoas com esta atitude.

Será descrito também um passo-a-passo de como deve ser desenvolvido um código fonte que seja bem documentado e de fácil manutenibilidade, utilizando padrões de código que facilitam em que todos os desenvolvedores tenham um conhecimento assertivo sobre seus códigos.

**2 CONCEITOS SOBRE O DESENVOLVEDOR DE SOFTWARE**

Com o grande aumento da tecnologia, as empresas necessitam cada vez mais de alguém capacitado que possa trazer um resultado com qualidade em questão de software, neste contexto, seguem os principais conceitos e funções do profissional que desenvolve softwares, com um breve histórico da profissão antes.

**2.1** Histórico da profissão

Há certa dificuldade para se chegar a um consenso sobre quem foi de fato o primeiro programador, apesar de uma ser profissão predominante masculina, muitos livros definem Augusta Ada Byron como sendo a primeira programadora. Conhecida atualmente por Ada Lovelace, ela nasceu em 1815, foi matemática e escritora inglesa, sendo que a partir do ano de 1842 ela escreveu o primeiro algoritmo para ser processado por uma máquina, a máquina analítica de Charles Babbage, conforme afirma Santino (2015):

Augusta Ada King, nascida em 1815 com o nome Augusta Ada Byron, conhecida como Condessa de Lovelace ajudou o colega, *Charles Babbage*, no desenvolvimento da primeira máquina de cálculo, além de ser responsável pelo algoritmo que poderia ser usado para calcular funções matemáticas. O trabalho de Ada Lovelace permitiria que a máquina calculasse os números de *Bernouilli*.

Ainda conforme Santino (2015), o dia 13 de outubro é utilizado para homenageá-la, esta data é reconhecida internacionalmente como "*Ada Lovelace Day*". Após Ada, houve várias outras pessoas que contribuíram na área de programação para que ela pudesse tornar-se o que é hoje. Seguem exemplos de importantes nomes para a tecnologia da informação, retirados do artigo de Gusmão (2016):

Grace Murray Hopper (1906 - 1992), trabalhou como analista de sistemas para Marinha dos Estados Unidos, além de criar a já extinta linguagem de programação chamada Flow-Matic que serviu como base para a criação do COBOL. Hopper é citada no artigo como "essencial para a finalização da linguagem COBOL";

Donald Knuth (1938), é o autor do livro The Art of Computer Programming (A Arte da Programação de Computador), é conhecido também como "pai da análise de algoritmos" por sua contribuição área da programação;

Vint Cerf (1943) foi em 2005 vice-presidente da Google, ainda é citado como "um dos grandes responsáveis pelo desenvolvimento da tecnologia TCP/IP" que é um dos alicerces para a internet e conexões à rede;

Linus Torvalds (1969) aos 21 anos, já tinha cinco anos de experiência em programação (em C). É chamado de "pai do Linux" pois lançou seu projeto em 1991.

2.2 Funções do Desenvolvedor de Software

Um profissional que atua como programador, desenvolvedor ou codificador de software tem como principais responsabilidades escrever, desenvolver ou fazer a manutenção do software. Ou seja, dentro da organização de uma empresa, os programadores são responsáveis apenas pelo desenvolvimento do software que lhes é passado por engenheiros e analistas de sistemas. Alguns desenvolvedores ainda preferem a atividade autônoma para desenvolvimento de softwares particulares, tendo que realizar toda a organização do projeto.

Ainda de acordo com o site Guia da Carreira (s.d.), o programador de software, tem como função principal: "criar uma série de instruções lógicas com o objetivo de executar determinadas sequências de ações. Estes comandos precisam estar escritos em modelos específicos, que são chamados de linguagens de programação." A partir deste conceito o programador realiza suas atividades profissionais.

**2.3** Classificações e ramificações dos Programadores

A profissão de desenvolvimento de software possui uma vasta gama de ramificações, podendo ser dividida em várias categorias. Seguem algumas dessas categorias, sugeridas por Silva (2013):

**2.3.1** Programador Desktop

Desenvolvedor de programas ou aplicações destinadas a funcionar em um computador ou rede de computadores, que normalmente utiliza um sistema operacional como: Windows, Linux ou MAC.

**2.3.2** Programador Web

Responsáveis por desenvolver websites ou sistemas baseados em web, que são configurados para funcionarem em rede e acessível em qualquer lugar.

**2.3.3** Programador Mobile

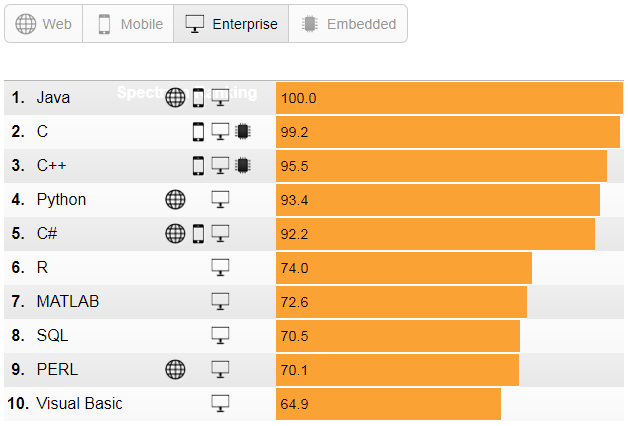
Profissional que desenvolve aplicações ou aplicativos para dispositivos móveis, normalmente utilizando uma das plataformas: Android, IOS, Windows Phone ou Java.

**2.3.4** Programador de Banco de Dados

Trabalha com a criação e o gerenciamento de banco de dados.

**2.4** Linguagens de Programação Mais Utilizadas

São muitas as linguagens de programação que podem ser utilizadas na criação de um programa ou software, cabe ao desenvolvedor e a empresa a escolha da mais adequada ao projeto. Segue abaixo uma pesquisa realizada pelo site spectrum.ieee com o ranking das linguagens mais utilizadas em 2014, para o ramo empresarial:



**2.5** Requisitos para se tornar um Programador

Ao contrário do que muitos pensam, muitas empresas não exigem diploma de nível superior para contratar o programador, apesar de que uma faculdade superior pode ser o diferencial para o candidato. Mas os principais requisitos para quem quer entrar nesse meio é "saber programar" (Adam Silva, 2013; Strauss, s.d.), ter um bom raciocínio lógico se possível ter conhecimento de mais de uma linguagem de programação, conhecimento na língua inglesa, ter o hábito de ler, ser esforçado e disciplinado. Após treinar essas habilidades, é recomendado buscar certificações e depois experiências, seja dentro ou fora de empresas.

**2.6** Atual mercado de trabalho

Apesar da área da informática estar em constante renovação e ser uma área ampla com muitas oportunidades há também grande concorrência por parte de empresas e profissionais, que devem estar constantemente atualizando-se conforme o mercado, para que não percam oportunidades valiosas. Segue abaixo o quadro com a média salarial de algumas profissões de TI conforme a pesquisa realizada por Ferrer (2012).

|  |  |
| --- | --- |
| **CARGO** | **MÉDIA SALARIAL** |
| Auxiliar de Programação | R$ 882,33 |
| Assistente de Programação | R$ 1.263,53 |
| Trainee de Programação | R$ 1.213,38 |
| Programador Júnior | R$ 1.634,99 |
| Analista de Programação Júnior | R$ 2.169,52 |
| Programador Pleno | R$ 2.649,11 |
| Analista de Programação Pleno | R$ 3.595,95 |
| Programador Sênior | R$ 3.505,39 |
| Analista de Programação Sênior | R$ 5.763,07 |

QUADRO 1 - Cargo e Média Salarial

FONTE: Ferrer (2012)

**3 PRINCIPAIS CONDUTAS ÉTICAS DE TI**

O conceito ou o significado de ética pode ser muito amplo e difícil para ser concretizado em palavras, ainda mais quando a mesma é aplicada em profissões, pois nem todas são regulamentadas como o caso de profissionais de desenvolvimento de software. Sendo assim, nesses casos cabe a cada empresa criar seu código de ética e moral para a organização.

Partindo do conceito informado por Vignado (2015) de que, a palavra ética pode ser definida como um "conjunto de regras e preceitos de ordem valorativa e moral de um indivíduo, de um grupo social ou de uma sociedade", grande número de empresas desenvolvedoras de software adota práticas similares no seu código ético ou de conduta muitas vezes o aplicando a mais de uma profissão de TI. Abaixo seguem as praticas ou regras mais aplicadas com sua breve descrição, sendo estas informações retiradas do site Análise de Sistemas Unopar (2013) que foi baseado em códigos da ACM:

* Público: Referente a relação com o cliente final, ou seja, agir de forma "coerente" e correta para com o cliente;
* Cliente e Empregador: Prática caracterizada por "agir de uma maneira que é no melhor interesse de seus clientes e empregadores e de acordo com o interesse público";
* Produto: Referente a garantia de entrega do melhor produto possível, ou seja "assegurar que seus produtos e modificações relacionadas atendam aos mais altos padrões profissionais possíveis";
* Julgamento: Afirma que os profissionais deverão manter "integridade e independência em seu julgamento profissional";
* Gestão: Os profissionais "devem assinar e promover uma abordagem ética para a gestão de desenvolvimento de software e manutenção";
* Profissão: A regra aplica-se aos desenvolvedores, pois "devem avançar a integridade e reputação da profissão de acordo com o interesse público";
* Colegas: Refere-se quanto a relação com os colegas, afirmando que os profissionais devem ser "justos e apoiar os seus colegas";
* Self: Afirma que os profissionais de TI "devem participar na aprendizagem ao longo da vida em relação à prática de sua profissão e devem promover uma abordagem ética à prática da profissão."

Os exemplos citados são apenas alguns das muitas práticas aplicadas por empresas, sendo que cada uma das regras possui inúmeros requisitos internos, e é neste contexto que se encaixa os engenheiros de software para auxiliar nessa organização de regras e condutas além de acompanhar todo o processo de desenvolvimento do software a ser criado.

**3.1** Casos Reais de Profissionais Antiéticos

Mesmo com todos os códigos e precauções que as empresas e profissionais capacitados tomam, ainda há muitos casos antiéticos seja por informações vazadas, disseminação de vírus, alterações irregulares no código fonte, entre outros. Seguem alguns exemplos de falta de ética profissional, que vieram a público, de acordo com Silva (s.d.):

Continental Can

Uma empresa de Connecticut que desenvolveu uma base de dados de pessoal, na qual incluiu todos os seus empregados. Entremeado entre os dados típicos de pessoal, a empresa incluiu um campo (denominado a propósito red flag), que sinalizava quando a aposentadoria estava se aproximando, ou quando o indivíduo já estava habilitado a requerer a pensão.

Durante toda a década de 80, a 'facilidade' (e bota aspas nisso) esteve ativa, e sempre que ela era sinalizada para alguém, a empresa o despedia, mesmo após décadas de serviços leais. Em 1991, uma corte federal em Newark, NJ, reconheceu o direito de ex-empregados, por demissão injusta, e determinou o pagamento de indenizações que chegaram ao montante de US$ 445 milhões.

Desenvolvimento de um sistema contrariando dispositivos legais e de segurança

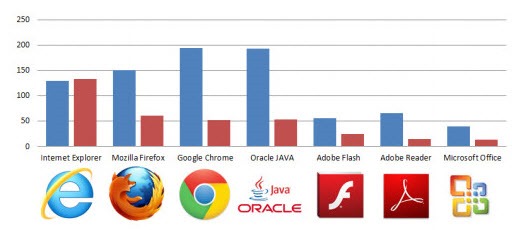
Um gerente de programação recebeu ordens para desenvolver um conjunto de programas que poderiam evitar os controles contábeis normais na firma de seu empregador. Foi-lhe explicado que o propósito era apenas testar algumas novas funções do negócio.

Ele protestou ao seu gerente sênior, mas lhe foi dito que os perigos de evitar os controles haviam sido avaliados e havia sido tomada a decisão de prosseguir da forma planejada. O gerente implementou os programas.

Como visto, estes casos infelizmente são verdadeiros, mas os profissionais e as empresas devem empenhar-se em aplicar as regras éticas, para que acontecimentos, que possam prejudicam o ambiente de trabalho, nunca mais ocorram.

**4 SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARES**

Ao se desenvolver um software é necessário que o profissional atente-se a alguns detalhes quanto a segurança do mesmo pois atualmente as questões de segurança atingiram uma complexidade maior com possíveis ataques de pessoas mal-intencionadas a partir do projeto do software a ser criado. A maneira mais eficaz de desenvolver um software seguro é incorporar a segurança desde o início do desenvolvimento de software caso contrário, a remoção as vulnerabilidades, no futuro poderá aumentar o custo do projeto significativamente. Abaixo encontra-se um gráfico com as vulnerabilidades encontradas em alguns aplicativos para computador, onde a cor azul representa as vulnerabilidades em 2013 e as barras vermelhas representam as vulnerabilidades do primeiro trimestre de 2014. Essa pesquisa foi retirada de um relatório da Bromium Security (2014).



Como previsto, os softwares browsers que são a "porta de entrada" para a internet, são os aplicativos com maiores vulnerabilidades e possuem maiores chances de receberem ataques de usuários e hackers mau-intencionados. Apesar da grande quantidade de riscos que há ao desenvolver um aplicativo, seja ele utilizado para uma conexão com a rede ou não, há algumas técnicas ou práticas que podem ser aplicadas no desenvolvimento de um software que auxiliam na segurança e organização do mesmo, como por exemplo as citadas por Molin (2011):

4.1 Gerenciamento de código fonte  
 Uma ferramenta para gerenciamento de código fonte permite organizar a interação entre desenvolvedores, garantir a integridade e possibilitar o gerenciamento de versões do código, evitando equívocos quanto a versão colocada em produção.

4.2 Realização de testes  
 Utilização de teste desde pequenos trechos de código por vez (como os testes unitários) até práticas que buscam uma avaliação geral do software.

4.3 Gerenciamento de correção de bugs  
 A utilização de uma ferramenta de Bug Tracking permite manter registro das falhas encontradas no sistema e facilitar a comunicação entre os envolvidos na identificação e correção de bugs.

4.4 Utilização de processo de integração contínua  
 Essa prática visa garantir a qualidade no software desenvolvido, Automatizando verificações no processo de build da ferramenta, garantindo assim, que seja possível gerar um novo release com o mínimo possível de bugs.

4.5 Documentação do software e da arquitetura que o suporta  
 Uma documentação clara da arquitetura e código fonte ajuda a aumentar a qualidade do software desenvolvido. Uma documentação clara, objetiva e bem estruturada é de fundamental importância para que o software possa ser expandido de forma sustentável e segura.

4.6 Utilização de padrões de código seguro e checklists  
 Defina padrões de códigos seguros, boas práticas de acordo com a linguagem adotada e ambiente definido pela organização. Crie checklists para verificar as principais ações durante o desenvolvimento e revisão de segurança do software.

**5 CONSIDERAÇÕES FINAIS**

**REFERÊNCIAS**

CÓDIGO DE ÉTICA PARA PROFISSIONAIS DE TI. 02 jun 2013. Análise de Sistemas Unopar. Disponível em: <http://analisedesistemasunopar.blogspot.com.br/2013/06/codigo-de-etica-para-profissionais-de-ti.html/>. Acesso em: 14 nov. 2017.

FERRER, Rafael. Os salários de 150 profissões na área de TI. 16 abr 2012. Exame Abril. Disponível em: <http://exame.abril.com.br/carreira/pesquisa-mostra-a-media-salarial-brasileira-em-ti/>. Acesso em: 14 nov. 2017.

GUSMÃO, Gustavo. Os 22 programadores mais importantes da história. 13 set 2016. Exame Abril. Disponível em: <http://exame.abril.com.br/tecnologia/os-22-programadores-mais-influentes-da-historia-recente/>. Acesso em: 17 nov. 2017.

INFORMÁTICA. (S.D.). GUIA DA CARREIRA.

Disponível em: <http://www.guiadacarreira.com.br/profissao/informatica/>. Acesso em: 14 nov. 2017.

SILVA, Adam. Programador – quanto ganha e o que faz?. 03 mar. 2013. Adam Silva.

Disponível em: <http://www.adamsilva.com.br/profissoes/programador/>. Acesso em: 19 nov. 2017.

SANTINO, Renato. Conheça Ada Lovelace, a 1ª programadora da história. 13 out 2015. Olhar Digital. Disponível em: <https://olhardigital.uol.com.br/noticia/conheca-ada-lovelace-a-1-programadora-da-historia/40718/>. Acesso em: 19 nov. 2017.

SILVA, Marcelo Kapczinski da. A ética do profissional da informática. (s.d.). Inf Ufes.

Disponível em: <https://inf.ufes.br/~fvarejao/cs/etica.htm>. Acesso em: 14 nov. 2017.

STRAUSS, Rodrigo. Como ser um bom programador. (s.d.). 1 Bit.

Disponível em: <http://www.1bit.com.br/content.1bit/bom\_programador/>. Acesso em: 19 nov. 2017.

VIGNADO, Alexandre. Valores e ética em gerenciamento de projetos. 03 out 2015. Administradores.

Disponível em: <http://www.administradores.com.br/mobile/artigos/carreira/valores-e-etica-em-gerenciamento-de-projetos/90769/>. Acesso em: 18 nov. 2017.

MOLIN, Cássio Mazzochi. Boas praticas para desenvolvimento de softwares seguros. [???] 2011. Micreiros.

Artigo do Seminário de Segurança em Desenvolvimento de Sistemas 2011-2

Disponível em: <<http://micreiros.com/boas-praticas-para-desenvolvimento-de-softwares-seguros/>>. Acesso em: 18 nov. 2017.

Stephen Cass, Nick Diakopoulos and Joshua J. Romero. Interactive: The Top Programming Languages. 03 jul 2014. Spectrum.ieee.

Disponível em: <https://spectrum.ieee.org/static/interactive-the-top-programming-languages>

Acesso em: 19 nov. 2017.

PPLWARE. Internet Explorer é o browser mais… inseguro. 27 JUL 2014

Disponível em: <https://pplware.sapo.pt/microsoft/windows/internet-explorer-e-o-browser-maisinseguro>

Acesso em: 14 nov. 2017.