



**CENTRO UNIVERSITÁRIO ANHANGUERA DE OSASCO –
UNIAN**

CURSO GESTÃO EM TI

PROJETO DE SOFTWARE

TRABALHO ACADÊMICO

OSASCO - SP

2020

PROJETO DE SOFTWARE

Anderson Silva Moura
Elían Vieira Costa
Leticia Lorrana Gonçalves Monteiro
Rafael Chaves de Oliveira
Thayna Martorini de Carvalho
Victor Santos Ferreira

Trabalho acadêmico da Faculdade Anhanguera de Osasco, apresentado como requisito para obtenção do título de Gestão em Tecnologia da Informação.

Orientador: Profº Ronaldo Barbato

Osasco

2020

PROJETO DE SOFTWARE. Dissertação apresentada a Faculdade Anhanguera de Osasco, para obtenção do título de Gestão em Tecnologia da Informação.

Aprovado em:

Banca Examinadora

Prof.: _____
Instituição: _____
Julgamento: _____
Assinatura: _____

Prof.: _____
Instituição: _____
Julgamento: _____
Assinatura: _____

Prof.: _____
Instituição: _____
Julgamento: _____
Assinatura: _____

RESUMO

Com o desenvolvimento das evoluções tecnológicas e o mercado cada vez mais competitivo, as empresas de software necessitam de pessoas com aptidão para melhor desempenho de suas atividades e assim agilizar seus processos internos. Diante disto, o trabalho vai estar explicando como um programador realiza o software e a forma de desenvolvê-lo.

ABSTRACT

With the development of technological developments and an increasingly competitive market, software companies need people with the skills to better perform their activities and thus streamline their internal processes. Given this, the work will be explaining how a programmer makes the software and how to develop it.

SUMARIO

1. INTRODUÇÃO.....	
2. OBJETIVO GERAL.....	
3. OBJETIVOS ESPECIFICOS.....	
4. JUSTIFICATIVA.....	
5. DESENVOLVIMENTO.....	
6. MODELO CONCEITUAL (IMAGENS).....	
7. MODELO LÓGICO (IMAGENS).....	
8. MODELO FISICO (CÓDIGO SQL)	
9. DICIONÁRIO DE DADOS.....	
10.UML-DIAGRAMA DE ESTADOS.....	

1. Introdução

O que é um software? Software é uma sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas. Também pode ser definido como os programas que comandam o funcionamento de um computador.

Em um computador, o software é classificado como a parte lógica cuja função é fornecer instruções para o hardware. O hardware é toda a parte física que constitui o computador, por exemplo, a CPU, a memória e os dispositivos de entrada e saída. O software é constituído por todos os programas que existem para um referido sistema, quer sejam produzidos pelo próprio utente ou pelo fabricante do computador.

E neste trabalho vou estar mostrando como projeta um software para uma empresa, por cada etapa, com isso você tem que saber o que seu cliente quer em seu software, os requisitos e os métodos do desenvolvimento que deverá ter.

2. Objetivo Geral

- Explicar como desenvolve um software.

3. Objetivo Especifico

- Mostrar passo a passo o que um programador necessita para criar um software a uma determinada empresa.
- Quais as formas de desenvolver um software.

4. Justificativa

Segundo Blaschek (2002, p.1):

Aos requisitos estão associados os principais problemas do desenvolvimento de software. Requisitos que não refletem as reais necessidades dos usuários, incompletos e/ou inconsistentes, mudanças em requisitos já acordados e a dificuldade para conseguir um entendimento comum entre usuários e desenvolvedores são as principais dificuldades relatadas, provocando retrabalho, atrasos no cronograma, custos ultrapassados e a insatisfação dos clientes e usuários de software.

Com isso, um software mal gerenciado pode causar transtornos como: implementações indevidas, retrabalho, e dúvidas dos envolvidos.

5. Desenvolvimento

Programador de Software

O programador tem a função de codificar e testar sistemas, nisso aplicam-se as atividades de manutenção dos sistemas e programas já desenvolvidos, realizando correções necessárias e pontuais. Ele cria programas desde o projeto até a finalização e rodagem do sistema, fazendo depuração e testes de programas.

Fara parte do seu dia a dia falar com o usuário e passar as informações. Ele terá que falar com o cliente de uma forma que ele entenda o que deseja como: qual objetivo do projeto; sobre os dados necessários para implantação; será um aplicativo, site ou sistema web? qual perfil de quem vai usar o projeto; forma de exibição das informações; quais processos serão automatizados.

Estes dados serão criados em uma linguagem mais formal, sendo necessário gerar documentos com concordância do projeto que serão anexados ao contrato.

Um programador de aplicativo tem várias atividades, algumas delas são: desenvolvimento, correções, deploys e suporte.

Desenvolvimento: normalmente começa com uma tarefa, onde é necessário criar o produto final, atuando com multitarefas como interface, criação de base de dados, realizando testes, front-end, etc.

Correções: como desenvolvedor você precisa encontrar prováveis erros de sistema como um código, um problema operacional, uso da memória e outras atualizações de software ou mesmo algo bem mais complicado, apresentando soluções.

Deploys: nesta parte o programador libera para o cliente sua criação, envia informações ao servidor, instala APIs ou outros dados dentro do sistema. É bastante comum um software ter que funcionar em vários aparelhos ou com integração entre eles, para evitar futuros problemas, várias aplicações podem ser feitas

Suporte: sempre que requisitado, será preciso que o desenvolvedor esteja presente para auxílio ao usuário, lembrando que este é uma pessoa leiga, então usar uma linguagem que seja de fácil compreensão será de grande ajuda.

Como desenvolver um software

Há dois rumos básicos no desenvolvimento de software: Desenvolvimento de Aplicativos e Desenvolvimento de Sistemas. O primeiro mantém seu foco na criação de programas que suprem a necessidade dos usuários. Eles podem variar de aplicativos para dispositivos móveis à produção avançada de videogames e a softwares de contabilidade empresarial. O segundo, por sua vez, está focado na criação e manutenção de sistemas operacionais com o desenvolvimento de ciclos de vida. O desenvolvimento de sistemas frequentemente envolve operabilidade de rede e segurança de dados.

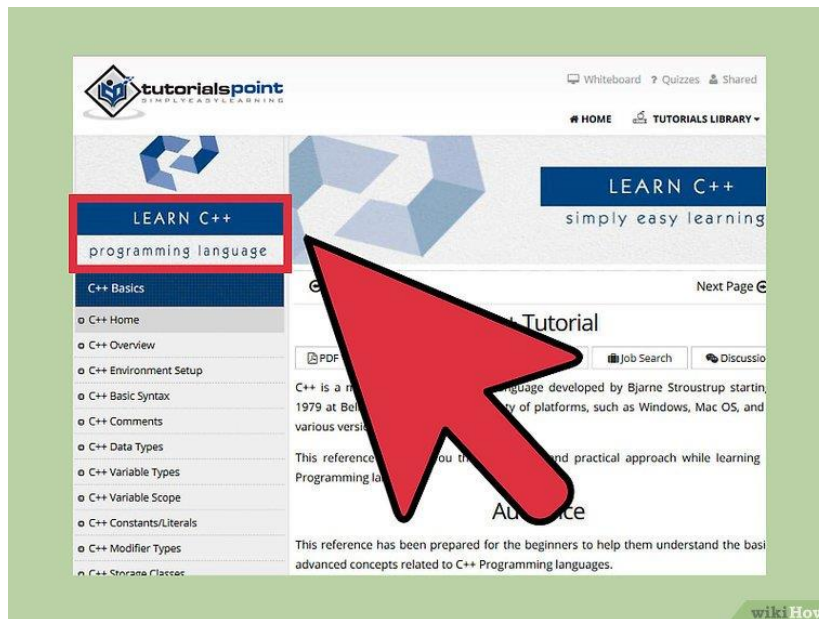
	<u>IT 131 - PROGRAMMING LOGIC</u>
	<u>IT 153 - WEB DEVELOPMENT</u>
	<u>IT 282 - DYNAMIC WEB DEVELOPMENT</u>
	<u>IT 284 - WEB PROGRAMMING</u>
	<u>CS 161 - COBOL PROGRAMMING</u>
	<u>CS 230 - EVENT-DRIVEN PROGRAMMING</u>
	<u>IT 280 - ADVANCED SYSTEMS DEVELOPMENT</u>
	<u>IT 221 - DATABASE DESIGN USING SQL</u>
	<u>CS 251 - OBJECT-ORIENTED PROGRAMMING</u>
	<u>CS 281 - ADVANCED OBJECT-ORIENTED PROGRAMMING</u>
	<u>IT 295 - SPECIAL ADVANCED PROJECTS</u>
	<u>CS 105 - FOUNDATIONS OF INFORMATION TECHNOLOGY</u>
	<u>CIS 110 - BUSINESS APPLICATIONS FOR MICROCOMPUTERS</u>

wikiHow

Fonte: https://www.wikihow.com/images_en/thumb/2/24/Develop-Software-Step-1-Version-2.jpg/v4-760px-Develop-Software-Step-1-Version-2.jpg

Qualquer pessoa pode ter ideias, mas um desenvolvedor será capaz de transformar as suas em algo tangível. Mesmo que você somente queira trabalhar nos aspectos visuais de um software, deve ter certa familiaridade com a codificação e ser capaz de criar protótipos básicos. Há uma imensa variedade de linguagens de programação que você pode ensinar a si mesmo. Algumas das mais úteis e importantes incluem:

- C — Uma das mais antigas linguagens ainda em uso, sendo a base para a maioria das outras aqui listadas. C é usada para desenvolver programas simples, e trabalha de modo próximo ao hardware do computador.
- C++ — é a versão orientada a objetos da C, sendo a mais popular linguagem de programação no mundo. Programas como Chrome, Firefox, Photoshop e muitos outros são todos criados com a C++. Ela é também uma linguagem muito popular na criação de videogames. Desenvolvedores C++ estão quase sempre em grande demanda.
- Java — uma evolução da linguagem C++, sendo usada devido à sua facilidade de portabilidade. Quase qualquer sistema pode rodar uma Java Virtual Machine, permitindo nele ser executado o software Java. Ela é amplamente usada em videogames e softwares de negócios, e muitas pessoas a recomendam como uma linguagem essencial.
- C# — Linguagem baseada em Windows que faz parte do framework .NET da Microsoft. Ela está proximamente relacionada à Java e à C++, e, se você aprender Java inicialmente, é possível transferir-se rapidamente para a C#. Essa linguagem é especialmente útil para desenvolvedores trabalhando com softwares para Windows ou Windows Phone.
- Objective-C — Outra prima da linguagem C, especificamente projetada para sistemas Apple, e desfruta de imensa popularidade em aplicativos para iPhone e iPad. Ela é uma linguagem para ser aprendida como freelancer.
- Python — Linguagem incrivelmente fácil de se aprender (uma das mais fáceis). A Python se especializa no desenvolvimento para web.
- PHP — Não se trata exatamente do desenvolvimento de software, mas a PHP é essencial se tem interesse em ingressar no desenvolvimento de software. Sempre há montes de trabalho para desenvolvedores PHP, embora não se trate de algo tão lucrativo como o desenvolvimento de software.



Fonte: https://www.wikihow.com/images_en/thumb/3/3b/Develop-Software-Step-2-Version-2.jpg/v4-760px-Develop-Software-Step-2-Version-2.jpg

Conclusão

Uma empresa de desenvolvimento de sistemas deve saber avaliar o melhor para si e a correta utilização dos seus recursos alocando as competências conforme a atividade a ser desenvolvida. É importante ressaltar que, as empresas que adotam a postura de gerenciamento de projetos devem estar em um nível de maturidade organizacional elevado, ou, pelo menos, consciente dessa necessidade. Isto fará com que os problemas que certamente existirão possam ser minimizados e resolvidos facilmente. O software de Projetos atua como facilitador na otimização da estrutura e processos, mas é fundamental que aconteça uma crescente evolução das técnicas, metodologias e modelos envolvidos visando o constante aumento da qualidade dos produtos

A replicabilidade de experimentos e análises estatísticas é um fator importante na produção de conhecimento científico porque quando um experimento ou análise de dados é replicado temos maior segurança de que os procedimentos seguidos foram bem descritos, podendo ser devidamente avaliados por outros cientistas, o que reduz o risco dos resultados encontrados serem apenas erros metodológicos. Com software proprietário, a comunidade científica até pode replicar resultados utilizando o mesmo software, mas sem acesso ao código fonte dos softwares utilizados, a inspeção é incompleta. Software livre, por definição, tem o código fonte disponível para inspeção por qualquer um que tenha competência para examiná-lo, e muitos cientistas estão suficientemente bem preparados para fazer isso. Essa

disponibilidade do código fonte aumenta a profundidade da inspeção possível dos procedimentos seguidos em experimentos e análises de dados. Software livre, portanto, é mais consonante com o princípio de replicabilidade do que software proprietário. É claro que um nível mais profundo de replicabilidade pode não ser justificativa suficiente para o uso de um software livre se sua qualidade for inferior à de softwares proprietários. Mas, como vimos, software livre usualmente tem alta qualidade, superando suas alternativas proprietárias, principalmente nos casos em que os usuários são numerosos e competentes como programadores. Esse é justamente o caso dos cientistas de muitas áreas. Especificamente, muitos estatísticos e alguns cientistas sociais são também competentes como programadores. Mesmo os que aprenderam a programar apenas na própria linguagem R contribuem usando o programa e alguns de seus milhares de pacotes adicionais e, eventualmente, fazendo sugestões de melhoria aos desenvolvedores do software. Por essas razões, o R se tornou um bom exemplo de software livre de alta qualidade. A comunidade de usuários do R é numerosa, e crescente, enfrentando, entretanto, alguns obstáculos a uma maior difusão entre cientistas sociais brasileiros. Nossos cursos de graduação e de pós-graduação na área de ciências sociais são, em sua maioria, deficientes no ensino de metodologia quantitativa (SOARES, 2005). Os estudantes são pouco ou nada treinados em projetos de pesquisa em que o número de casos analisados é grande (centenas ou milhares de casos) e em que o objetivo da pesquisa é fazer generalizações. Em poucos cursos, há ênfase no uso de ferramentas estatísticas. Os cientistas sociais brasileiros estão mais habituados a trabalhar com textos e discursos do que com ferramentas lógicas e matemáticas. Assim, o esforço inicial para usar um software como o R, que exige habilidade de programador, é maior para estudantes de ciências sociais do que para estudantes de outras áreas. Além disso, os softwares proprietários concorrentes do R investem em propaganda e em técnicas comerciais de fidelização. É comum, por exemplo, a distribuição de licenças de uso para estudantes a baixo custo ou mesmo gratuitas. Após aprenderem a fazer suas análises de dados com um software, o pesquisador precisa estar muito motivado a usar o R para deixar temporariamente a fase produtiva em que se encontra e voltar à fase de aprendizado. Mas, no longo prazo, o esforço compensa. E, na verdade, o esforço não precisa ser tão grande como se pensa. Existe muito material de livre acesso na internet ensinando a dar os primeiros passos no R, e existem pacotes do R — como o Deducer e Rcmdr — que criam menus semelhantes aos de softwares proprietários, exibindo exemplos de código que facilitam o aprendizado ao tornar menor o número de comandos a serem memorizados pelos iniciantes.

Referência

GUEDES, G. T.A **UML 2: uma abordagem prática. 2. Ed. São Paulo: Novatec, 2011. p. 26.**

O PROCESSO de Desenvolvimento de Software: Detalhes sobre a fase de construção.
Disponível em: <[HTTP://www.dsc.ufcg.edu.br](http://www.dsc.ufcg.edu.br)