FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Análise e Desenvolvimento de Sistemas

Thiago Henrique Felix

SIGE – SISTEMA INTEGRADO PARA O GERENCIAMENTO ESCOLAR

Americana, SP

2017

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Análise e Desenvolvimento de Sistemas

Thiago Henrique Felix

SIGE – SISTEMA INTEGRADO PARA O GERENCIAMENTO ESCOLAR

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Análise e Desenvolvimento de Sistemas, sob a orientação do (a) Prof.(a)
Titulação maior do mesmo (Esp. / Me. / Dr.)
Nome completo do(a) professor(a).

Área de concentração: Engenharia de Software

Americana, S. P.

2017

- Ficha Catalográfica –
Solicitada previamente e enviada
por e-mail pela Biblioteca.
Que deve ser impressa
neste formato, **no verso**
da folha acima.

Thiago Henrique Felix

SIGE – SISTEMA INTEGRADO PARA O GERENCIAMENTO ESCOLAR

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Analise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.
Área de concentração: colocar apenas uma (01) área temática identificada na ficha de inscrição do curso.

Americana, de mês de defesa da banca de 9999.

Banca Examinadora:

_____ Nome completo do orientador (Presidente)
Maior titulação do orientador
Instituição de atuação

_____
Nome completo do membro da banca (Membro)
Maior titulação
Instituição de atuação

_____
Nome completo do membro da banca (Membro)
Maior titulação
Instituição de atuação

AGRADECIMENTOS

DEDICATÓRIA

Aos meus Pais, e todos os amigos que me acompanharam nesta caminha de três anos rumo a graduação de tecnólogo.

RESUMO

O objetivo deste trabalho é o desenvolvimento de um sistema integrado para o gerenciamento escolar, o sistema será composto por módulos sendo estes programas específicos de determinada área. O foco deste sistema é informatizar a educação, saindo do tradicional papel para uma plataforma confiável.

Palavras Chave: Sistema; Gerenciamento; Diagrama;

ABSTRACT

Keywords:

SUMÁRIO

LISTA DE FIGURAS E DE TABELAS

INTRODUÇÃO

A tecnologia ao longo das últimas décadas mudou o estilo de vida das pessoas, facilitado suas vidas, softwares de bancos possibilitam transações bancarias em poucos segundos através de aparelhos celulares, compras do mês realizadas em qualquer lugar através da internet, além da conexão de pessoas em diferentes lugares do mundo. Mas a evolução da tecnologia não traz apenas benefícios para as pessoas, mas também para empresas e o setor público com sistemas de informação capazes de facilitar o gerenciamento.

Mesmo com a tecnologia trazendo tantos avanços e possibilidades como os sistemas de informação computacionais, existem comércios, empresas, setores públicos, entre outros, que gerenciam seus negócios no papel.

É importante se atentar que sistemas de informação computacionais não são baratos, sua criação gera um custo e muitas vezes pequenos comércios tem receio em investir ou até mesmo há uma inexistência de recursos para o investimento, e é ai que entra o software livre, softwares criados por uma comunidade de diversos programadores ao redor do mundo e normalmente licenciados com uma licença de código aberto (*Open Source*) ou software livre (*Free Software*), a diferença entre esses dois termos ficará mais clara ao desenvolver do trabalho.

Este modelo de software possibilita a entrada de pequenos negócios ou organizações de qualquer tipo que ainda possuem dúvidas sobre a melhoria que os sistemas de informação podem oferecer para seus negócios, já que normalmente possuem uma grande diferença para com os softwares proprietários que é em relação ao seu preço.

O objetivo deste trabalho é o desenvolvimento de um sistema integrado para o gerenciamento escolar que utilizará a licença GPL (GNU *General Public License*), sendo útil tanto no setor público quanto para o setor privado, tendo como foco na implementação de sistemas de informação em entidades escolares que ainda duvidam deste potencial sem qualquer custo financeiro.

O sistema será composto por módulos sendo estes programas específicos de determinada área. O sistema irá informatizar a educação, através de um software livre.

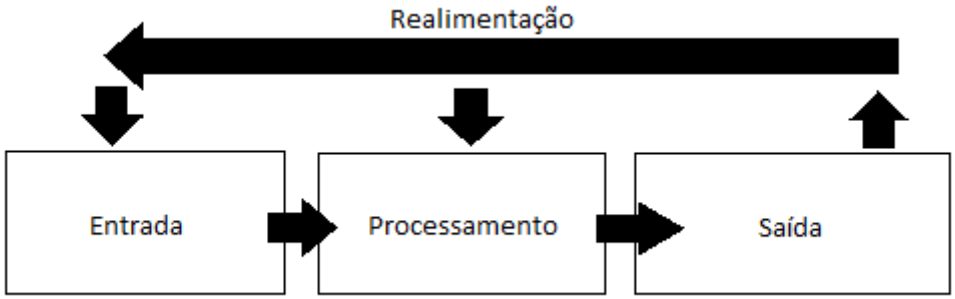
Para criação desta plataforma estudantil irei utilizar a linguagem de programação web PHP em sua versão 7 juntamente com o banco de dados Firebird. Como ferramenta de desenvolvimento irei utilizar o editor de texto Atom juntamente com o gerenciador de banco de dados o Flamerobin, o controle de versão do software será feito através do Github, sendo todas estas ferramentas de uso gratuito.

O presente trabalho é estruturado em quatro capítulos diferentes, o primeiro capitulo tem como objetivo situar o leitor com os presentes termos e conceitos aqui referenciados como a definição de sistemas de informação, sistemas de informação gerencial, software livre, licença GPL, diferenças entre software livre e código aberto e por fim conhecer um pouco da linguagem utilizada no desenvolvimento. O segundo capitulo irá focar em todos os requisitos do sistema, ou seja, os principais objetivos para com este sistema sendo eles funcionais ou não funcionais. No terceiro capitulo será explicado e demonstrado toda a documentação do software sendo estes resultantes da engenharia de software e do banco de dados. E finalmente no ultimo capitulo será possível ver o sistema completo juntamente com seu manual de uso.

1. SISTEMAS DE INFORMAÇÃO

Um sistema de informação (SI) é um conjunto de componentes inter-relacionados para um determinado fim, estes componentes são: coleta de dados (entrada), manipulação de dados (processamento), armazenamento e retorno de dados (saída) e a realimentação de dados corretivo (mecanismo de realimentação), este mecanismo de realimentação é o elemento que leva as organizações a alcançarem seus objetivos, que com a análise dos dados de saída, realimenta o sistema com novos dados, como por exemplo um sistema de informação de uma loja de roupas, que consta que em uma determinada região um tipo de roupa vende em um número mais acelerado que em relação a outras regiões, analisando esta informação é possível concluir que este produto é essencial nesta região, sendo assim o estoque deve ser aumentado.

Figura 1: Componentes de um sistema de informação



Fonte: Próprio autor

Nos sistemas de informação o componente de entrada é onde os dados serão captados como por exemplo em um sistema de vendas, é necessário que se insira as informações do produto antes que fique disponível para o cliente, é na entrada onde estes dados serão inseridos ou em um sistema de rotas que é necessário o local antes do processamento para a melhor rota.

O componente de processamento é onde se converte os dados em informações úteis com determinado proposito, para tal feito pode ser necessário a realização de cálculos, comparação de dados ou armazenamento o processamento é a parte crucial de um sistema de informações. Neste componente é onde um sistema que determina preço de um produto faz todos os cálculos necessários adicionando todo o custo de fabricação do produto, lucro além de impostos, ou um sistema de vendas online onde é feito cálculos dos produtos mais vendidos, mais procurados, menos procurados gerando um relatório para tomadas de decisão. Após todo o processamento a informação gerada pode ser armazenado para possíveis usos no futuro ou mesmo comparada com outras informações auxiliando para tomada de decisões.

O componente de saída é o resultado do processamento, ou seja, informações que auxiliam na toma de decisões, estas informações podem ser em formatos de relatórios, ou informações para gerentes, bancos, acionistas, dentre outros. As saídas de um sistema pode ser a entrada de outros sistemas, sendo estas informações processadas novamente e gerando novas informações úteis.

O componente de realimentação é a informações originada do sistema, que é utilizada para mudar a entrada de dados ou processamento do sistema. Esta mudança seria uma atualização no sistema com a análise na informação de saída, são feitos novos parâmetros no sistema tanto na entrada de dados quanto no processamento de dados, um exemplo disto seria em um sistema de vendas que registra todo produto vendido e faz uma baixa no estoque, em sua saída no final do mês seria constado que a baixa no estoque de determinados produtos ficou abaixo de zero, o sistema não deveria realizar a baixa de estoque quanto o produto chega-se a zero com a análise da saída do sistema, é feita a realimentação atualizando a entrada de dados, não aceitando produtos com o estoque abaixo de um.

1.0.1 SISTEMAS DE INFORMAÇÃO GERENCIAL

Stair e Reynolds (2009, p.19) definem: “Um sistema de informação gerencial (MIS – *managiment information system*) é um conjunto organizado de pessoas, procedimentos, softwares, banco de dados e equipamentos que fornecem informações rotineiras aos gerentes”. Este tipo de sistema de informação chamado de MIS, tem foco em trazer relatórios periódicos aos gerentes e tomadores de decisão tendo foco na eficiência organizacional englobando várias áreas organizacionais como marketing, produção, recursos humanos entre outras áreas.

Os MIS foram desenvolvidos nos anos de 1960 gerando relatórios periódicos para determinadas áreas organizacionais, com o passar do tempo os MIS proliferaram para todos os níveis organizacionais gerenciáveis.

1.1 SOFTWARE LIVRE

Na década de 1940 no surgimento dos primeiros projetos de computadores como ENIAC *(Eletronic Numerical Integratir and Computer*) ainda não havia o conceito de software, todo foco de pesquisa e desenvolvimento era no hardware, nas próximas décadas o hardware ganha espaço no cenário comercial ao mesmo tempo os pesquisadores começam a criar software para estes equipamentos com o intuito de torna-los úteis.

Nesta época os softwares eram criados por pesquisadores e o código era compartilhado entre universidades, como ainda não existia internet o código era enviado por correio e assim eram criados em comunidade, com o objetivo de aprender e melhorar o código.

Como o passar do tempo ficou evidente o potencial comercial do software com seus benefícios as grandes empresas, era um produto comercial incrível é aí que o cenário começa a mudar, um exemplo desta mudança foi o sistema operacional UNIX que inicialmente foi criado por pesquisadores, foi um marco importante na história da computação sendo vários softwares importantes baseados nele.

UNIX em sua criação tinha o código fonte divulgado entre universidades com o objetivo de se aprender e melhorar sendo uma importante descoberta cientifica, mas com o tempo os softwares começaram a ser desenvolvidos em larga escala com as empresas obrigando os programadores a assinarem acordos de não revelação, os programadores não poderiam divulgar o código fonte sendo vendido somente o código binário como um produto.

Richard Matthew Stallman nos anos 70 trabalhava no laboratório de IA (Inteligência Artificial) do MIT, nesta época a comunidade vivia seu auge, universidades compartilhavam entre si códigos fontes, o que na época era natural.

Stallman havia tido problemas com uma impressora nos laboratórios do MIT com atolamento de papeis, mas com uma mudança no código fonte da impressora foi possível avisar aos usuários que do problema ocorrido, a alteração não resolvia o

problema, mas evitava que seus usuários ficassem esperando por um longo período de tempo.

O laboratório de IA do MIT certa vez ganhou uma impressora de presente da Xerox, Stallman procurou o código fonte da impressora como sempre fazia, mas com a pesquisa percebeu que somente era disponibilizado o código binário da impressora, ainda pesquisando Stallman encontrou um dos programadores do software da impressora e foi até ele, ao pedir o código fonte da impressora o programador o avisou que ele assinou um acordo de não divulgação não podendo divulgar o código fonte da impressora.

Foi aí que Stallman percebeu que o mundo no qual ele havia ajudado a crescer estava mudando o código fonte não era mais compartilhado, vários de seus amigos estavam assinando acordos de não revelação, ele percebeu que com este cenário crescente as empresas poderiam ciar o monopólio do software já que mesmo pagando não se teria mais o direito de modificar o código. Então Stallman tomou a decisão, deixar o MIT para ajudar no desenvolvimento do software livre.

Stallman iniciou em 1984 o projeto GNU (GNU is not Unix) com o objetivo de criar um sistema operacional assim com UNIX juntamente com a comunidade e de código aberto disponível gratuitamente para todas as pessoas. No ano seguinte em 1985 Stallman criou a Fundação de Software Livre (FSF – Free Software Foundetion), uma organização sem fins lucrativos a partir disto foi criada a licença GPL (*General Public License*), esta licença garante que o software permaneça livre para sempre, sendo assim uma empresa não pode incluir o código com esta licença e vender como proprietário.

1.2 LICENÇA PÚBLICA GERAL GNU (GPL)

1.3 ENGENHARIA DE SOFTWARE

A engenharia de software é um ramo da engenharia cujo foco é o desenvolvimento dentro de custos adequados de sistemas de software de alta qualidade (SOMMERVILLE, 2007, p.3), o termo surgiu em 1968 em uma conferência na qual tinha-se o objetivo de discutir o que então era chamado de “Crise de software”, este termo surgiu após o lançamento de um novo hardware de computadores baseado em circuitos integrados, com isso veio a oportunidade de se criar software grandes que poderiam suprir as necessidades de grandes empresas, o problema era criar tais softwares sem métodos ou conceitos para garantir a qualidade do software dentro do prazo de desenvolvimento, e com isso surgiu a crise de software com projetos que superavam seus custos e prazos. Era necessário a criação de técnicas e conceitos para criação de grandes softwares, algumas das técnicas ainda existem até hoje enquanto novas são criadas a cada dia.

1.3.1 PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE

Com o avanço rapido do hardware ao longo dos anos, veio a possibilidade de criação de grandes sistemas de informação mas devido a grande complexidade em seu desenvolvimento foi necessario a criação de que na engenharia de software é chamado de processo de software. "Um processo de software é um conjunto de atividades que leva à produção de um produto de software" (SOMMERVILE,2007,p. 42), ou seja é um conjunto de atividades necessaria para o desenvolvimento seguro do software, as atividades fundamentais independente do processo de software são:

Especificação de software: é a especificação de todas as funcionalides do software e suas restrições, normalmente em softwares comerciais isso é definido juntamente com o cliente.

Projeto e implantação de software: Essas atividades tem como objetivo principal assegurar que as especificações do software sejam produzidas corretamente.

Validação do software: O software deve fazer o que foi especificado, nesta atividade essas especificações devem ser validadas no software.

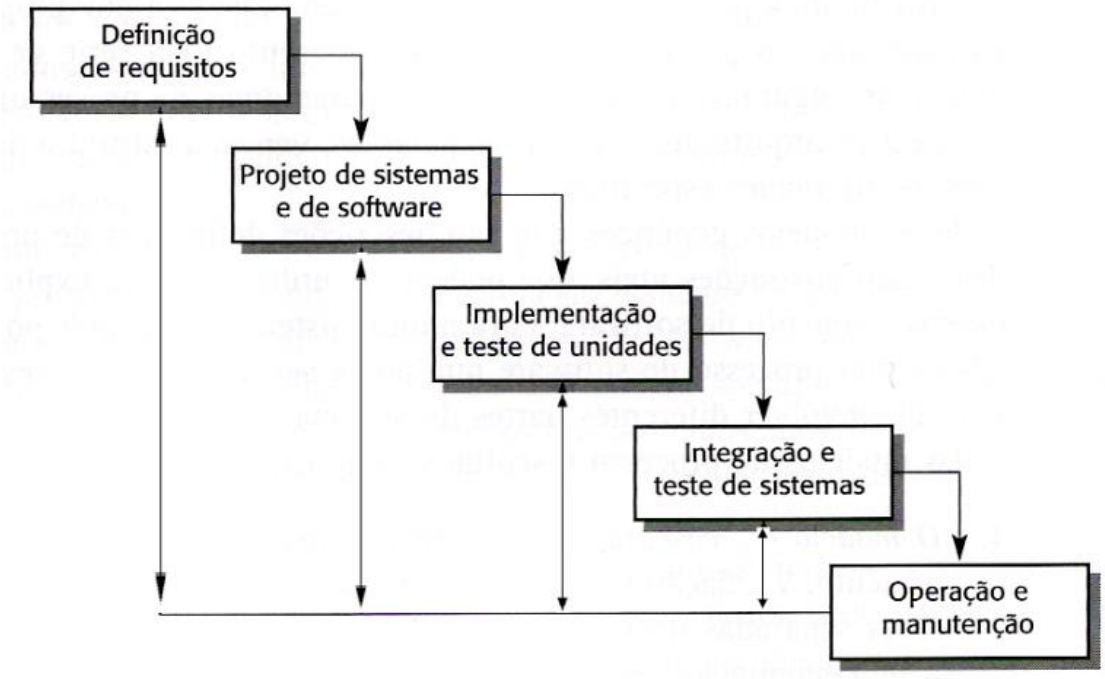
Evolução do software: As regras de negocios mudam com o tempo e os softwares devem acompanhar sendo assim necessario a evolução.

Não existe um processo de software correto para todos os projetos, o processo deve ser escolhido analisando o projeto, com um projeto complexo e com requisitos imutaveis o modelo cascata é muito recomendado, mas para projetos com poucos requisitos e em constante mudança processos de desenvovimento ágeis como scrum ou xtreaming programming são os mais recomendados, a seguir será demonstrado os processos mais conhecidos sendo os já citados incluidos.

1.3.1.1. CASCATA

O modelo de processo de software cascata é um modelo tradicional, sendo utilizado somente em projetos especificos, nos quais cada atividade deve ser devidamente especificada para assim ir para a próxima. Na modelo cascata uma atividade somente pode ser iniciada se sua antecessora estiver terminada, devido a esta regra que este processo de software tem o nome de cascara demonstrada na Figura 2.

Figura 2: Modelo de processo de software cascata

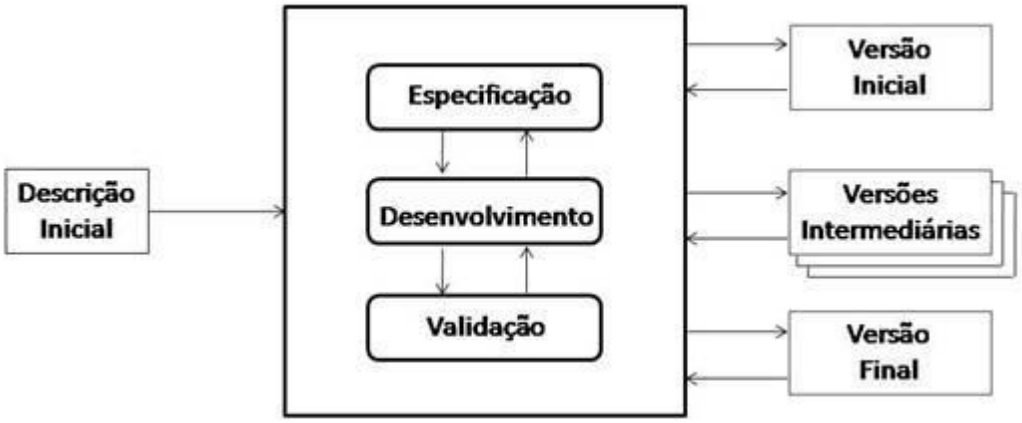


Fonte: http://www.ebah.pt/content/ABAAAfBgcAI/sistema-sanepar-na-linguagem-delph

1.3.1.2. EVOLUCIONARIO

O modelo de processo de software evolucionário pode ser utilizado em projetos nos quais os requisitos são abstratos onde nem mesmo os clientes sabem ao certo o que quererem, neste modelo primeiramente é desenvolvido uma versão inicial do sistema e baseado nos comentários do cliente, o sistema é modificado até sua versão final onde os objetivos do cliente são alcançados. Existem dois tipos de desenvolvimento evolucionário o primeiro é o desenvolvimento exploratório no qual o objetivo é trabalhar juntamente com o cliente explorando os requisitos até o desenvolvimento de um sistema final, o segundo chama-se prototipação *throwaway* (descartável) o objetivo deste é entender os requisitos focando-se na experimentação de requisitos mal compreendidos.

Figura 3: Desenvolvimento evolucionário



Fonte: Ian Sommerville, 2007, p.46.

A grande vantagem dessa versão é que sempre há especificação de requisitos sendo assim é muito mais fácil desenvolver um sistema ideal para o cliente, porém a documentação do sistema pode ser caótica.

1.3.1.3. INCREMENTAL

1.3.1.4. ESPIRAL

1.3.1.5. MÉTODOS ÁGEIS

1.4 FERRAMENTAS DE DESENVOLVIMENTO

1.4.1 PHP

PHP é uma linguagem de programação criada em 1994 por Rasmus Lerdof, neste tempo a linguagem era apenas scripts para páginas dinâmicas que Rasmus utilizava para monitorar seu currículo online, mais tarde em 1995 foi lançado uma versão do PHP chamada de PHP/FI (*Personal Home Pages/Forms Interpreter*). Neste mesmo ano Rasmus liberou o código fonte do PHP para o público permitindo que qualquer um pode-se alterar os scripts. Com isso diversas pessoas tiveram a oportunidade de ajudar no desenvolvimento ajudando a achar bugs e os corrigi-los, ou seja, ajudar a linguagem a melhorar.

Figura 4: PHP logo

Fonte: https://pt.wikipedia.org/wiki/PHP#/media/File:PHP-logo.svg

Em 1997 PHP já era usado em uma grande parte da internet, sendo lançada no mesmo ano a segunda versão do PHP. Em uma pesquisa de Netcaft em 1998 mostrou que por volta de 60.000 domínios relataram ter uma parte em PHP.

Terceira versão do PHP foi escrita em 1997 quando dois jovens de uma Universidade de Israel chamados de Andi Gutmans e Zeev Suraski de Tel Aviv estavam desenvolvendo uma aplicação de eCommerce para um projeto da Universidade, mas a segunda versão do PHP não era suficiente foi aí então que os

começaram a reescrever o interpretador. Discutindo com Rasmus online os três decidiram criar uma nova linguagem na qual foi chamada simplesmente de ‘PHP’ tornando o significado um acrônimo recursivo - PHP: Hypertext Preprocessor. Deste então Andi Gutmans e Zeev Suraski vem trabalhando em novas versões do PHP chegando até a atual PHP 7.

1.4.2 MYSQL

1.4.3.1 CODEIGNITER

2 LEVANTAMENTO DE REQUISITOS

Os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007, p. 79), para o desenvolvimento de um sistema integrado para o gerenciamento escolar, será documentado os requisitos funcionais e não funcionais

REFERÊNCIAS BIBLIOGRÁFICAS

OLIVA, Alexandre et al (Org.). **A revolução do Software Livre.** Manaus: Comunidade Sol - Software Livre, 2009. p. 1-58

GUEDES, Gilleanes T.A. **UML2**: uma abordagem prática. São Paulo/SP: Novatec, 2009.

STAIR, Ralph M.; REYNOLDS, George W. **Princípios de Sistemas de Informação.** Trad**.** Harue Avritscher. 9ª. ed. São Paulo/SP: Cengage Learning, 2011. p. 1-20.

OGLIO, Pablo Dall'. **PHP:** programando com orientação a objetos. São Paulo/SP: Novatec, 2007.