

PROPOSTA DE DESENVOLVIMENTO DE UM APLICATIVO DE GESTÃO DE PROVAS

RODRIGO QUEIROZ DE FARIA
rodrigoqfaria@hotmail.com
FACEC

FABIO MACHADO DE OLIVEIRA
fabiomachado.prof@gmail.com
FACEC

FLAVIO PAVESI SIMAO
fpsimao@ifes.edu.br
IFES

Resumo: Este trabalho tem com o objetivo desenvolver um projeto de desenvolvimento de um software de gestão de provas, utilizando tecnologia da informação para ajudar as instituições de ensino a elaborar e gerir suas provas ajudando ter um controle das provas já aplicadas demonstrando que o uso da tecnologia pode ajudar uma instituição a ter um melhor desempenho, podendo assim, oferecer muitas melhorias a fim de gastar menos tempo e obter uma maior gestão desses resultados. Foram utilizados métodos de pesquisa bibliográfica e experimental, abordando técnicas de engenharia de software com modelo de prototipação.

Palavras Chave: Gestão acadêmica - Tecnologia - Desempenho - -

1. INTRODUÇÃO

A tecnologia da informação proporciona a grandes e pequenas empresas ganhos de desempenho, lucros e confiabilidade de dados, uma vez que a tecnologia nos ajuda a organizar, sempre utilizando os recursos tecnológicos disponíveis para cada caso, uma organização totalmente gerenciada por recursos tecnológicos pode prover relatórios e estatísticas.

Tais recursos tecnológicos utilizam algum meio de armazenamento de dados que trabalha, gravando os dados em tabelas para futuramente eventuais consultas possibilitando a recuperação das informações armazenadas, uma das funções que ele agrega é nos permitir relacionas dados possibilitando o cruzamento dessas informações, além de garantir restrições de segurança a dados.

Requisitos funcionais dos sistemas podem ser expressos em formas de diagramas para facilitar o entendimento dos usuários do sistema.

Existe uma grande necessidade do uso de sistemas de apoio em muitas áreas do mercado e no meio acadêmico, o uso da tecnologia é uma das melhores maneiras de melhorar os processos neste meio.

Processos que são repetitivos como elaborações de provas, podem ser auxiliados por sistemas tecnológicos com objetivo de facilitar e gerenciar todo processo tornando o processo mais simples.

Sistema tem como proposta ser uma ferramenta de auxílio para professores e instituições, de forma a melhorar construção das provas e gestão dos dados. Através do sistema professores podem utilizar esse tempo para fazer outras atividades, como por exemplo, elaborar projetos acadêmicos para o corpo discente, além da possibilidade da geração de relatórios estatísticos disponibilizado pelo sistema.

2. REVISÃO BIBLIOGRÁFICA

2.1. INFORMAÇÃO

Segundo Gordon e Gordon (2006) gestores de informações podem tomar proveito da informação e torna-la conhecimento. Conhecimento é o entendimento sobre o objeto alvo. Um dado é capaz de gerar a informação que pode ser transformada em conhecimento. Sabedoria é a capacidade de uma pessoa utilizar-se de seus conhecimentos, gerados da informação para um proposito.

Gordon e Gordon (2006) cita que podemos encontrar os tipos de informação:

- Informação com um recurso: usa informação para proporcionar desempenho, ou melhor, aproveitamento de seus bens;
- Informação como um ativo: podem visualizar o conteúdo de suas informações e utiliza-la como um investimento, utilizando como vantagem completa;
- Informação como produto: as empresas podem vender a informação como produto, uma vez que depois de trabalhada por sistemas de informações pode agregar valores.



2.2. SISTEMAS DE INFORMAÇÃO

De acordo com Laudon e Laudon (2007) podemos definir como sistema de informação um conjunto de componente que inter-relacionam que recuperam informação e processam, armazenam e distribuem essas informações de forma organizada.

Alcides e França (2008) afirmam que qualquer sistema que utilize de recursos tecnológicos ou que manipule informações pode ser considerado um sistema de informação, quando uma informação que é trabalhada sistemas informação, possibilitam a geração de cenários e simulações que auxiliam os processos de tomada de decisões na empresa, sistemas de informações que não se propõem atender esses mínimos objetivos, eles se tende a se tornam desnecessários.

Alcides e França (2008) concluem que o ciclo de vida de um sistema de informação abrange:

- **Concepção:** trata-se do nascimento do sistema, onde o projeto do sistema é elaborado;
- **Construção:** trata-se da execução do sistema, complementando o desenvolvimento dos documentos a análise do sistema se for necessário à programação do sistema;
- **Implantação:** disponibilizar o sistema para o cliente, após o termino do período de testes e toda devida documentação do sistema de informação elaborado;
- **Implementações:** consiste em melhorar o sistema de informação que já foi implantando, seja novas funções específicas;
- **Maturidade:** consiste no período onde todos os requisitos funcionais do sistema de informação satisfazem as eventuais necessidades que o cliente necessita em suas operações;
- **Declínio:** período onde existe uma dificuldade de manter o sistema funcionando corretamente seja por impossibilidade de agregar novas funções sem aferir outras funcionalidades ou ate mesmo insatisfação do cliente, onde o uso do sistema de informação passa começar a se tornar desnecessário;
- **Manutenção:** consiste no período onde são elaboradas manutenções corretivas, seja por necessidade legal, seja por necessidade de correção de alguma eventual falha do sistema ou até mesmo para atender alguma nova necessidade do cliente visando prolongar a vida do sistema;
- **Morte:** período onde o sistema não é mais viável para a empresa onde o uso dele se tornar desnecessário, também podemos dizer que é período onde o sistema é descontinuado.

2.3. TECNOLOGIA DA INFORMAÇÃO

Conforme Daniel, Giulio e Eduardo (2003) até o ano de 1930 a evolução dos computadores era vastamente venta, a necessidade do uso de computadores ainda não era bem aceita, a partir do ano de 1930 a necessidade do uso computação eletrônica foi entendida.

Para Alcides e França (2008) a tecnologia da informação é uma ferramenta, que não pode ser descartada, o uso dela é indispensável para uma organização e normalmente o termo



tecnologia tende a assustar pessoas que não estão devidamente acostumadas ou familiarizadas com este meio tecnológico.

2.4. .NET FRAMEWORK

Lima e Reis (2002) citam que o *.NET Framework* é uma nova plataforma de desenvolvimento de software, totalmente desenvolvida pela *Microsoft* e totalmente inovadora que tem como objetivo a produção de softwares de várias plataformas diferentes como para aplicações web, aplicações desktop e aplicações para dispositivos móveis.

Lotar (2007) pronuncia em sua obra que o *.NET Framework* possui um gerenciador de código, que se chama CLR que permite que todo código seja compilado em tempo de execução, assim o computador tem um melhor gerenciamento de memória evitando gastos de memórias necessários e o desenvolvedor não precisa se preocupar em alocar e liberar memória, evitando estouros de memórias, que podem prejudicar a maturidade e a qualidade do software.

Lima e Reis (2002) indicam que *.NET Framework* é uma plataforma de desenvolvimento que engloba várias outras linguagens de programação, compiladores, modelo de objetos com objetivo de se tornar uma única ferramenta completa de desenvolvimento. As linguagens suportadas na plataforma *.NET* são C#, C++, Visual Basic e Jscript.

2.5. LINGUAGEM C#

Stellman e Greene (2008) revelam que a C# é uma linguagem de programação poderosa e robusta que permite um grande ganho de desempenho no desenvolvimento, pois sua estrutura se propôs a ajudar o desenvolvedor, pois possibilita um grande ambiente automatizado para se fazer um simples botão funcionar, não é preciso fazer um código enorme e complexo, pois o ambiente de desenvolvimento gera o código automaticamente, assim somente nos preocupamos com que realmente precisamos a contrario de outras linguagens de programação existentes no mercado.

2.6. MECANISMO DE BANCO DE DADOS

Teorey, Lightstone, Nadeau (2007) dizem que um componente básico em qualquer sistema de informação que manipule dados é o arquivo de banco de dados, ele é responsável por guardar informações de forma estruturada e organizada além de garantir as devidas restrições de segurança, um banco de dados é um objeto completo ele é capaz de armazenar dados, inter-relaciona-os.

Teorey, Lightstone e Nadeau (2007) advertem que o modelo entidade relacional é formado por três classes: Entidades, que são formadas por objetos de dados onde os dados são guardados; Relacionamentos, que representam as associações reais entre as entidades do banco de dados, e; Atributos, que são os campos contidos nas entidades dos sistemas de banco de dados.

Dhingra e Swanson (2007) observam que o *SQL Server Compact 2005* é um banco de dados leve, bastante adequado para dispositivos com baixo poder de processamento de dados, ele possui mecanismo de otimização em consulta a dados.



2.7. ENGENHARIA DE SOFTWARE

2.7.1. Análise de Requisitos

Bezerra (2007) repara que um processo de desenvolvimento de software é uma atividade muito complexa, um processo de desenvolvimento de software requer diversos componentes para que um processo chegue até o fim sem atrasos e muitos projetos de desenvolvimento acabam desperdiçando tempo e dinheiro. O projeto precisa definir todas as atividades necessárias, desenvolver, testar e manter o produto final de software.

Para Bezerra (2007) existem três tipos de requisitos:

- Requisitos funcionais: são funcionalidades disponíveis do sistema;
- Requisitos não funcionais: declaram às qualidades o sistema tem que possuir e estão relacionadas com suas funcionalidades;
- Requisitos normativos: são declarações de restrições impostas sobre o desenvolvimento do sistema.

Bezerra (2007) percebe que um documento de requisitos não deve conter informações sobre soluções técnicas que serão utilizadas no desenvolvimento o sistema, na realidade o documento de requisitos seve apenas para compreender o funcionamento do sistema antes de começar desenvolve-lo.

2.7.2. Fundamentos da UML

Nunes e O'Neill (2003) descreve que a *UML* é uma linguagem de modelação unificada, ela se utiliza da notação padrão para especificar, construir, visualizar e documentar os projetos de sistemas de informações. Por sua simplicidade e por seus conceitos o uso da UML facilita o desenvolvimento dos projetos de sistemas de informações.

2.7.3 Diagramas de Casos de Uso

Bezerra (2007) cita que a técnica de modelagem de casos de uso foi elaborada por um engenheiro de software sueco, chamado Ivar Jacobson, na década de 1990. Casos de uso é a especificação de uma sequencia de interações entre o sistema e seus agentes externos um caso de uso representa uma situação funcional de uma função do sistema sem expor seu funcionamento interno, cada caso de uso do sistema descreve de forma narrativa.

Nunes e O'Neill (2003) detalham que os componentes dos casos de uso são:

- Atores: uma das principais tarefas durante o desenvolvimento de um diagrama de use case é identificar os atores do sistema, os atores são representações de pessoas que fazem uma determinada operação ou ação;
- Uses Cases: os casos de uso são os processos do sistema também podem ser definidos como as regras de negocio do sistema;
- Relações (Include, Extends, Generalizações): os casos de uso podem se relacionar entre si a relação seja relacionamento entre atores com atores, casos de uso e casos de uso e casos de uso e atores;

2.7.4. Diagramas de Classe

Nunes e O'Neill (2003) atentam que um dos diagramas mais utilizados no desenvolvimento de software orientados a objeto sem duvidas é o digrama de classes. Objetivo do diagrama de classe é a discrição formal da estrutura dos objetos de dados de um



sistema. Cada objeto descreve sua identidade e seus devidos relacionamentos com outros objetos, atributos e métodos.

Nunes e O'Neill (2003, apud RUMBAUGH, 1991). Citado por lembra que o desenvolvimento de um modelo de classes é resultado de um processo na qual se identificam os objetos importantes que se pretende modelar, procura se descrever as propriedades atributos, o os métodos operações. Neste contexto as classes descrevem objetos que contem atributos e operações, servem para compreender o mundo real e tais informações importantes para o sistema.

Nunes e O'Neill (2003) dizem que o diagrama de classes devidamente compostos são formados por:

- Objeto: um objeto tem como objetivo incorporar um modelo de informação pode ser caracterizado como conjunto de propriedades, comportamentos, identidades.
- Propriedades: tem como objetivo definir o objeto em um conjunto de atributos, sendo que os valores estabelecem o estado do objeto atual.
- Comportamento: são operações quais os métodos do objeto podem efetuar.
- Identidade: principal objetivo é identificar o objeto como único conjunto de objeto parecido.

Uma classe representa de uma estrutura de abstração sobre conjunto de objetos que compartilham da mesma estrutura e comportamentos de característica similares, uma classe define seu comportamento seus objetos pelos seus métodos e possíveis estados desses objetos por seus atributos.

Tipo de dados. Cada atributo tem que ser identificado pelo seu tipo de dados, que tem como objetivo caracterizar o tipo de informação que deve conter mais, os tipos de dados estão relacionados a cada linguagem de programação utilizada que o sistema será desenvolvido, os tipos básicos em qualquer linguagem de programação são:

- Integer : representa um número inteiro;
- Long: representa um número inteiro de maior dimensão;
- Double: para números reais;
- String: representa texto;
- Date: para datas;
- Boolean: valor lógico que representa Verdade ou Falso.

Associações : as relações entre objetos são chamados de associações.

2.7.5. Prototipação

Daniel, Giulio e Eduardo (2003) visualizam que a prototipação consistem em reproduzir uma ideia, com objetivo de permitir que o usuário questione o produto de software e outros processos de análise de requisitos sejam executados com a visualização de parte do produto funcionando, sem usar tal complexidade que o sistema da prostra em pratica exijam.

Daniel, Giulio e Eduardo (2003) listam que as vantagens do uso da técnica de prototipação são:

- Menor tempo para o projeto entrar em produção;
- Usuário está presente em vários momentos evitando grandes retrabalhos;
- Produto de software agregando maior qualidade;
- Grande redução de custos desnecessários;

3. METODOLOGIA

A pesquisa desse trabalho divide-se em duas partes pesquisa bibliográfica e experimental, onde se necessita para desenvolver o produto, técnicas de engenharia de software, seguido pelo modelo de prototipação, também foram utilizadas pesquisas experimentais para produzir o produto de software, sendo por se tratar de um produto genérico, não podendo se prender a uma instituição.

Objetivo é desenvolver uma proposta de software para controle de provas, a proposta desenvolvida controla todo o processo, possibilitando ter todas as provas, alunos e questões catalogadas para eventuais necessidades.

Informações sobre as regras de negocio do sistema serão estudas e transpostas para uma análise de requisitos para posteriormente onde ela possibilitar inicio os demais diagramas e formando uma completa documentação.

Sistema será desenvolvido posteriormente em *C#* uma linguagem totalmente orientada a objetos aliada com mecanismo de banco de dados *SQL Server 2005 Compact* onde ambas são desenvolvida pela *Microsoft*.

4 RESULTADOS

Com as especificações de requisitos projeto de desenvolvimento de software de um sistema de gestão de provas, utilizou-se casos de uso, diagramas de classe, dicionário de dados e interfaces do usuário afim de construir e documentar um sistema de gestão de provas.

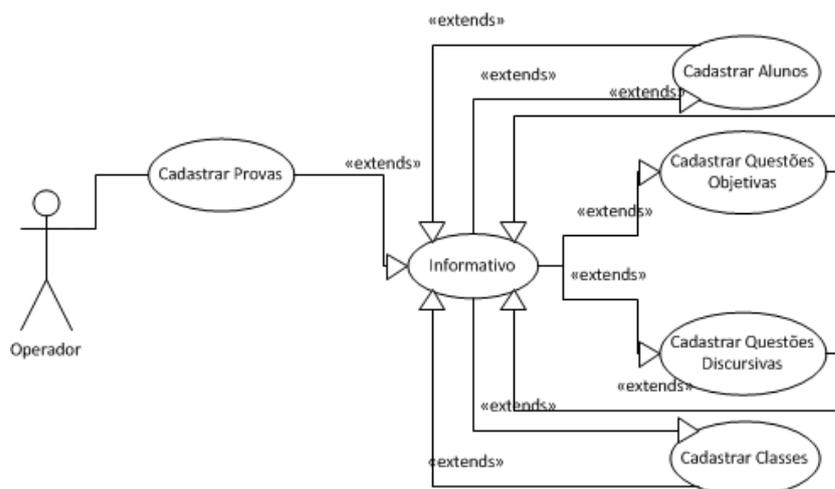
O objetivo específico incluiu:

- Automatizar o processo de elaboração de provas
- Obter um maior ganho de desempenho na hora de elaborar uma prova.
- Possibilitar a impressão de provas com cabeçalho pronto com nome do aluno, data e classe.
- Disponibilizar um histórico de alunos, provas e questões.
- Possibilitar uma maior confiabilidade das provas, além de minimizar erros.

Dentre os métodos utilizados para a elaboração do sistema, o diagrama de use case proporcionou uma amplo entendimento e visão do sistema a ser proposto. Como referencia da técnica utilizada, é apresentado abaixo um modelo utilizado para a função “cadastrar prova”, conforme segue abaixo:

- Cadastrar provas

Operador é responsável por cadastrar as provas.





DESCRIÇÃO DE CASO DE USO

Projeto: Prova Fácil

Nome do Caso de Uso: Cadastrar provas

Analista: Rodrigo Queiroz de Faria

Data: 01/10/2011

Descrição: Este caso de uso tem como objetivo demonstrar como será o processo de cadastro de provas.

Atores: Operador.

Pré-Condição: Não existe.

Pós-Condição: Cadastro da prova feito com sucesso.

Fluxo Principal:

1. Ator colocara nos campos os dados da questão objetiva e clica em salvar.

Extensões ou Fluxos Alternativos:

1. Ator colocara nos campos os dados da questão objetiva e clica em salvar.
2. Acontece alguém erro ao salvar o registro e será informado o motivo.
3. Pede para efetuar o cadastro de algum item que não foi cadastrado.

Como funções de saída, pode-se apresentar como exemplo a tela de impressão da prova, com a devida descrição da função, conforme abaixo:

Tela de impressão

Imprimir

Instituição:		NOME DA INSTITUIÇÃO	
Data:	01/01/0001	CodigoProva:	00000000
Anho:	00000001	Rodrigo Queiroz de Faria	
Classe:	000000ES	Sis2008-2011	

1)Em programação orientada a objetos, é correto afirmar que herança múltipla?
Resposta:

2)A implementação de herança múltipla em Java não é possível?
Resposta:

3) Ao se retornar um valor de função declarada como void, como ocorre entre as linhas de 24 a 29, produz-se erro de sintaxe.
3.1) Certo.
3.2) Errado.

4) No contexto de sistemas Web, qual a utilidade de uma sessão?

Imprimir

DESCRIÇÃO DO FORMULÁRIO

Projeto: Prova Fácil

Nome do Formulário: FrmImpressao

Analista: Rodrigo Queiroz de Faria



Data: 01/10/2011

Descrição: Objetivo da tela é gerar a imagem da prova e imprimir.

Campos de entrada: Não contem.

Campos de saída: Não contem.

O sistema ainda contempla as seguintes funções:

- Ficha de alteração e cadastro de alunos;
- Tela de listagem de alunos por turma;
- Ficha de alteração e cadastro de classe;
- Tela de listagem de classes;
- Ficha de alteração e cadastro de disciplina;
- Tela de listagem de disciplinas;
- Ficha de alteração e cadastro de questões discursivas;
- Tela de listagem de questões discursivas;
- Ficha de alteração e cadastro de questões objetivas;
- Tela de listagem de questões objetivas;
- Ficha de alteração e cadastro de respostas de questões objetivas;
- Tela de geração de prova, data e valor;
- Tela de geração da prova e seleção dos alunos;
- Tela de geração de prova e seleção de questões;
- Tela de geração de questões discursivas;
- Tela de geração de questões objetivas;
- Tela de geração dos alunos para impressão

5. CONCLUSÃO

Conclui-se que através do uso de sistemas de informações no processo de emissão de gestão de provas é possível melhorar e agilizar o processo. Em base dos estudos realizados é possível obter uma conclusão e podemos relatar que sistemas computadorizados são capazes de gerenciar todo o processo.

O ganho de desempenho na geração da prova é progressivo e vai se maximizando de acordo que o banco de dados de questões vai se enriquecendo de questões.

Por se tratar de um projeto de ciclo de desenvolvimento do software pode necessitar de melhorias ou implementações por se tratar de um projeto experimental.

6. REFERÊNCIAS

ALCIDES, Denis; FRANÇA, Aline. **Tecnologia da Informação Aplicada a Sistemas de Informação Empresariais**. São Paulo: Editora Atlas, 2008.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Campus, 2007.

GORDON, Judith R.; GORDON, Steven R. **Sistemas de Informação: Uma Abordagem Gerencial**. Rio de Janeiro: LTC, 2006.

DANIEL, Ricardo; GIULIO, Enrico; EDUARDO, Fernando. **Introdução à Ciência da Computação**. São Paulo: Thomson, 2003.

DENNIS, Alan; HALEY, Barbara. **Análise e Projetos de Sistemas**. Rio de Janeiro: LTC, 2005.



LAUDON, Kenneth C; LAUDON, Jane P. **Sistemas de Informação Gerenciais**. São Paulo: Pearson, 2007.

LIMA, Edwin; REIS, Eugênio. **C# e .net – Guia do Desenvolvedor**. Rio de Janeiro: Campus, 2002.

LOTAR, Alfredo. **Como Programar com ASP. NET e C#**. Rio de Janeiro: NovaTec, 2007.

DHINGRA, Prashant; SWANSON, Trent. **Microsoft SQL Server 2005 Compact Edition**. United States of America: Sams Publishing, 2008.

NORTON, Peter. **Introdução à Informática**. São Paulo: Pearson, 2007.

NUNES, Mauro; O'NEILL, Henrique. **Fundamental de UML**. Lisboa: FCA, 2003.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. Rio de Janeiro: Elsevier, 2006.

STELLMAN, Andrew; GREENE, Jennifer. **Use a Cabeça c#**. Rio de Janeiro: Alta Books, 2008.

TOEREY, T.; Lightstone; S.; Nadeau, T. **Database Modeling & Design: Logical Design**. San Francisco: Elsevier, 2005.