

Universidade Federal do Ceará
Campus Quixadá
Curso de Sistemas de Informação

**Ambiente Virtual de Aprendizagem para Auxiliar no
Processo de Ensino e Aprendizagem de Matemática**

Projeto de Pesquisa

Marciano Machado Saraiva

Orientador:
Prof. Msc. Samy Soares Passos de Sá

Quixadá
Julho, 2016

SUMÁRIO

1	INTRODUÇÃO	2
1.1	Objetivos do Trabalho	3
1.2	Divisão do Trabalho	3
2	REVISÃO BIBLIOGRÁFICA	4
2.1	Metodologias no Ensino da Matemática	4
2.1.1	Aula expositiva	4
2.1.2	Resolução de problemas	5
2.1.3	Modelagem Matemática	5
2.1.4	O Uso de Computadores	6
2.2	Ambientes Virtuais de Aprendizagem	7
2.3	Gamificação	9
2.4	Trabalhos Relacionados	10
2.4.1	<i>The one world schoolhouse: Education reimagined</i>	10
2.4.2	<i>ActiveMath: A generic and adaptive web-based learning environment</i>	11
2.4.3	<i>Duolingo: Learn a Language for Free while Helping to Translate the Web</i>	12
2.4.4	<i>Resolução de problemas em ambientes virtuais de aprendizagem num curso de licenciatura em matemática na modalidade a distância</i>	13
2.5	Considerações finais do capítulo	14
3	PROCEDIMENTOS METODOLÓGICOS	14
3.1	Definição do Processo	14
3.2	Levantamento e Análise de Requisitos	16
3.3	Projeto do Sistema	19
3.3.1	Arquitetura	19
3.3.2	Ferramentas	20
3.4	Implementação do Sistema	20
3.5	Verificação e Validação	21
3.6	Definição dos Conteúdo para o Sistema	21
3.7	Aplicação da Solução na Universidade Federal do Ceará - Campus Quixadá	22
3.8	Cronograma de Execução	22
4	RESULTADOS PRELIMINARES	23
	REFERÊNCIAS	28
	APÊNDICE A - DOCUMENTO DE REQUISITOS	29
	APÊNDICE B - DOCUMENTO DE PERSONAS	38
	APÊNDICE C - DOCUMENTO DE ARQUITETURA	42

1 INTRODUÇÃO

A Matemática, como ciência, tem uma relação muito especial com as novas tecnologias desde as calculadoras aos computadores, sistemas multimídia, e a Internet. No entanto, alguns professores costumam demorar a perceber como tirar proveito destas tecnologias como ferramentas de trabalho (PONTE, 1997).

À medida que a quantidade de recursos tecnológicos na sala de aula foram aumentando, tornou-se conveniente a criação de novas metodologias de ensino, especificamente na Educação Matemática. A busca por novas metodologias de ensino que fazem uso destes recursos busca fazer da matemática uma disciplina atraente e desvinculada do ensino tradicional que já se mostrou ineficiente (SILVA, 2009).

Grande parte dos alunos tem dificuldades em aprender matemática, e muitas vezes essas dificuldades ocorrem não pela falta de atenção ou por não gostar do conteúdo, mas por fatores mentais ou psicológicos que envolvem uma série de trabalhos e conceitos que precisam ser desenvolvidos (SÁ, 2015). Mas como auxiliar alunos com dificuldades na aprendizagem da matemática?

Em busca dessa resposta, diferentes sistemas de *softwares* foram desenvolvidos buscando servir à educação. Em 2006, Salman Khan fundou a Khan Academy, uma organização educacional que tem por objetivo oferecer exercícios, vídeos de instrução e um painel de aprendizado personalizado que habilita os estudantes a aprender no seu próprio ritmo dentro e fora da sala de aula (KHAN, 2012). A plataforma criada por Khan utiliza vídeo-aulas e resolução de problemas para ensinar seus alunos, permitindo que cada um aprenda de forma independente e no seu próprio ritmo.

Um outro trabalho também importante nessa área, é o de Melis et al. (2001). Os autores desenvolvem um AVA que permite aos alunos desfrutarem da experiência de estudar num curso gerado dinamicamente. Os conteúdos são representados num formato XML (BRAY et al., 1998) e armazenados numa base de conhecimento, onde são recuperados para gerar cursos individualmente de acordo com regras pedagógicas¹ (MELIS; SIEKMANN, 2004). Estes e outros trabalhos são apresentados com mais detalhes em trabalhos relacionados.

O presente trabalho apresenta, o projeto de um Ambiente Virtual de Aprendizagem (AVA) (VALENTINI; SOARES, 2010), para auxiliar estudantes no ensino e aprendizagem de conteúdos matemáticos. Este ambiente se propõe a servir como ferramenta para estudantes que buscam estudar fora do ambiente escolar e no seu próprio ritmo. O ambiente também dará suporte ao ensino e aprendizagem em sala de aula, auxiliando professores com informações relevantes sobre o andamento do aprendizado de cada um de

¹ São regras que determinam quando e quais ferramentas do sistema devem apresentadas e em que ordem.

seus alunos, além das dificuldades que os mesmos apresentam.

A ideia deste AVA surgiu quando professores das disciplinas de Matemática da Universidade Federal do Ceará observaram um grande número de reprovações e desistências nas disciplinas iniciais de matemática ao qual eles ministram. Segundo os professores, um dos fatores que pode ser o causador são as deficiências de formação em Matemática desde o ensino médio, e por isso falta aos alunos a base para compreender os novos assuntos.

Essa deficiência em Matemática pode ser constatada num estudo realizado pela Organização Para a Cooperação e Desenvolvimento Econômico (OCDE)(OECD, 2016). Esse estudo revelou que 67.1% dos alunos brasileiros ainda estão abaixo do nível 2 (os níveis são de 1 a 6) deixando o país em 58º lugar na escala do PISA², sendo que somente 0,8% dos estudantes brasileiros alcançaram os últimos patamares.

1.1 Objetivos do Trabalho

Este trabalho objetiva apresentar o projeto de um AVA para auxiliar no processo de ensino e aprendizagem de matemática por alunos dentro e fora da sala de aula, buscando contribuir com o processo de ensino e aprendizagem de matemática.

Como objetivos específicos para este trabalho, temos:

- Desenvolver as telas de administração do ambiente.
- Desenvolver as telas do uso pelos alunos aplicando técnicas de Gamificação.
- Verificar a influência e impactos provocados pelo AVA no aprendizado de alunos por meio de exames aplicados a cada aluno antes e após o uso do ambiente na Universidade Federal do Ceará.

1.2 Divisão do Trabalho

Fundamentando-se na problemática mencionada e tendo em vista o objeto de estudo, dividimos esta monografia em quatro capítulos. No capítulo inicial fizemos uma apresentação do tema e os objetivos do trabalho.

No Capítulo 2, destacaremos os aspectos teóricos sobre ensino e aprendizagem, assim como as tradicionais metodologias de ensino e as apoiadas por computador. Abordaremos também conceitos de gamificação e os trabalhos que servem de referência para os conceitos e idéias utilizadas no trabalho aqui desenvolvido.

No Capítulo 3, a concepção, construção e modelagem do sistema, apresentando o que o mesmo deve possuir e por que.

² O Programa Internacional de Avaliação de Estudantes (Pisa) é uma iniciativa de avaliação comparada, aplicada a estudantes na faixa dos 15 anos, idade em que se pressupõe o término da escolaridade básica obrigatória na maioria dos países e que visa melhorar as políticas e resultados educacionais.

No Capítulo 4, serão apresentados os resultados preliminares.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo aborda os seguintes temas: metodologias no ensino da matemática, conceito de ambientes virtuais de aprendizagem, o uso da gamificação aplicada em AVAs, e, por fim, o que está sendo desenvolvido por outros pesquisadores da área.

2.1 Metodologias no Ensino da Matemática

Ao longo da história, várias metodologias e abordagens matemáticas foram utilizadas visando a melhoria do ensino. Segundo Hammes (2003), algumas delas foram aula expositiva, resolução de problemas, modelagem matemática, e o uso de computadores. Nas seções a seguir, descreveremos cada uma delas.

2.1.1 Aula expositiva

Em uma aula expositiva, o professor comumente faz uma revisão da aula anterior, apresenta o novo conteúdo e passa aos alunos uma série de exercícios de fixação. Esse novo conteúdo é apresentado de forma oral ou escrita, sem levar em consideração o conhecimento prévio dos estudantes nem tempo para perguntas. Essa é, sem dúvida, uma das mais utilizadas e antigas metodologias existentes. Durante o século passado, aulas expositivas foram o único processo empregado em sala de aula pelos professores. Dessa forma, a aula expositiva pode ser considerada cansativa e desinteressante, já que o aluno não participa do processo de ensino (HAMMES, 2003).

Para ALENCAR (1996), essa abordagem possui diversos problemas. ALENCAR (1996) sugere que a escola tradicional não somente está desatualizada para atender às necessidades crescentes da sociedade contemporânea, como também apresenta algumas características que inibem o desenvolvimento do potencial de criação dos alunos:

- Destaca-se a incompetência, a ignorância e a incapacidade do aluno, deixando de assinalar os talentos e habilidades de cada um;
- O ensino voltado para o passado, em que se enfatiza a reprodução e a memorização do conhecimento;
- Desconsidera-se a imaginação e a fantasia como dimensões importantes da mente;
- Exercício de resposta única, em que se cultua o medo do erro e do fracasso;

- A obediência, dependência, passividade e conformismo são os traços mais cultivados;
- Descaso em cultivar uma visão otimista do futuro;
- As habilidades cognitivas são desenvolvidas de forma limitada;

Autores como Lopes e Veiga (1995), defendem que a aula expositiva “poderá ser transformada em uma atividade dinâmica, participativa e estimuladora do pensamento crítico do aluno”. Uma alternativa descrita por Lopes e Veiga (1995) é transformar a aula expositiva em uma aula expositiva dialógica. Essa forma de aula expositiva utiliza o diálogo entre professor e aluno para estabelecer uma relação de intercâmbio de conhecimentos e experiências (LOPES; VEIGA, 1995). De acordo com Freire e Guimarães (1982), o ensino dialógico se contrapõe ao ensino autoritário, transformando a sala de aula em ambiente propício à reelaboração e produção de conhecimentos.

2.1.2 Resolução de problemas

A resolução de problemas deve ser entendida como uma oportunidade para o aluno obter novos conhecimentos e não apenas conhecimentos prontos que fazem parte da nossa história. Ela ajuda, o aluno a desenvolver sua autonomia buscando as respostas para seus próprios questionamentos.

FOSSA e MENDES (1998) ressaltam que esta metodologia visa o desenvolvimento de habilidades metacognitivas, favorecendo a reflexão e o questionamento do aluno que aprende a pensar por si mesmo, levantando hipóteses, testando-as, tirando conclusões e até discutindo-as com os colegas.

Nessa metodologia, é importante que os problemas apresentem uma incógnita que necessite ser descoberta. Para resolvê-los, o aluno terá que inventar estratégias e gerar novas ideias. Segundo Dante (1991) é importante que o problema possa gerar muitos processos de pensamento, levantar muitas hipóteses e propiciar várias estratégias de solução. O pensar e o fazer criativo devem ser componentes fundamentais no processo de resolução de problemas.

2.1.3 Modelagem Matemática

De acordo com Bassanezi (2002, p. 16), “[...] a modelagem consiste na arte de transformar situações da realidade em problemas matemáticos cujas soluções devem ser interpretadas na linguagem do mundo real”. Nas metodologias anteriores, o processo de ensino é deflagrado pelo professor. Na Modelagem Matemática, o processo é compartilhado com o grupo de alunos, pois sua motivação advém do interesse pelo assunto.

A modelagem matemática é uma metodologia que busca proporcionar ao aluno uma visão prática do conhecimento teórico aprendido na sala de aula, através de problemas de ordem prática ou de natureza empírica (FOSSA; MENDES, 1998).

Burak (2004) destaca alguns aspectos importantes a cerca dos benefícios da utilização da modelagem matemática:

- Maior interesse do grupo, pois o fato do grupo escolher aquilo que gostaria de estudar, ter a oportunidade de se manifestar, de discutir e propor, desenvolve o interesse do grupo.
- Interação maior no processo de ensino e de aprendizagem, pois o grupo de alunos trabalha com aquilo que gostam e que apresenta significado, tornando-os co-responsáveis pela aprendizagem.

2.1.4 O Uso de Computadores

A aprendizagem mediada por computadores surgiu em 1960 na Universidade de Illinois com o projeto PLATO (Programmed Logic for Automatic Teaching Operations)(BITZER et al., 1961), que deu origem ao primeiro sistema de ensino assistido por computador, o qual permitia a criação e apresentação de materiais sobre gramática (passar um verbo para o passado, reescrever um substantivo no plural, etc.) com revisão automática. De acordo com Woolley (1994), PLATO apoiou inicialmente apenas uma única sala de aula com 20 alunos, até que em 1972, o sistema migrou para uma nova geração de *mainframes* que acabaria por apoiar milhares de terminais gráficos distribuídos em todo o mundo.

O principal fator motivador para a introdução do computador na educação, segundo Silva (2009), foi o surgimento no final do século XX de um conhecimento baseado em simulação, característico da cultura informática, o que fez com que o computador fosse visto como um recurso didático indispensável.

Os benefícios da utilização do computador como um instrumento de ensino e aprendizagem, de acordo com Almeida (2000, p.12), referem-se a sua utilização como “uma máquina que possibilita testar ideias ou hipóteses, que levam à criação de um mundo abstrato e simbólico, ao mesmo tempo em que permite introduzir diferentes formas de atuação e interação entre as pessoas”. Já a principal associação de professores de matemática dos Estados Unidos (NCTM), diz que “a tecnologia é essencial no ensino e na aprendizagem da Matemática” e “influencia a Matemática que é ensinada e melhora a aprendizagem dos alunos” permitindo que estes se concentrem “nas decisões a tomar, na reflexão, no raciocínio e na resolução de problemas”. (MELO, 2007, p.26).

Quando se fala no uso das Tecnologias da Informação e Comunicação (TIC) como

um mediador para o processo de ensino e aprendizagem, muitas vezes, acaba-se esquecendo do papel do professor nesse processo. Num mundo em que há uma grande variedade de formas de utilização das TIC para apoio a educação, cabe ao professor decidir como e quando utilizar essas tecnologias e quais são formas mais eficazes para sua utilização levando em consideração os conteúdos que serão ofertados. Para isso, o professor necessita mudar sua metodologia de ensino, e isso pode resultar em problemas, já que, assim como afirmam Bitner e Bitner (2002, p. 96):

“Adults do not change easily. Change of any kind brings about fear, anxiety, and concern. Using technology as a teaching and learning tool in the classroom does so to an even greater extent since it involves both changes in classroom procedures and the use of often-unfamiliar technologies. Those responsible for asking teachers to use technology in the curriculum should be aware that fears and concerns do exist.” (BITNER; BITNER, 2002, p. 96).

Os problemas com a inserção do computador na educação podem ser ainda maiores, tendo em vista que muito se cogita sobre seu uso no ensino ser a solução para muitos dos problemas da educação, sendo que muitos desses problemas não podem encontrar uma solução nas tecnologias digitais (SILVA, 2009). Um dos fatores que pode influenciar negativamente no processo de aprendizagem mediada por computador é o domínio do computador pelo aluno, tendo em vista que sua rapidez de evolução assim como sua própria complexidade, torna essa tarefa muito difícil de ser alcançada.

2.2 Ambientes Virtuais de Aprendizagem

O Surgimento de Ambientes Virtuais de aprendizagem (AVAs) deu-se logo após o surgimento da internet nos anos 90. Nessa mesma época, novas ferramentas e produtos foram desenvolvidas para explorar os benefícios que a rede mundial de computadores trouxe (OLEARY; RAMSDEN, 2002).

Para Valentini e Soares (2010), um Ambiente Virtual de Aprendizagem (AVA) é um espaço social, constituído de interações cognitivo-sociais sobre, ou em torno de, um objeto de conhecimento, no qual as pessoas interagem mediadas pela linguagem da hipermídia visando o processo de ensino-aprendizagem.

Geralmente, os AVAs possuem algumas características que os distinguem de outros tipos de sistemas de softwares. Segundo OLeary e Ramsden (2002), algumas dessas características são:

- a comunicação entre tutores e alunos - por exemplo, email, fórum de discussão e bata-papo virtual;
- a auto-avaliação e avaliação sumativa - por exemplo, avaliação de múltipla escolha com automatizada marcação e feedback imediato;

- entrega de recursos de aprendizagem e materiais - por exemplo, através do fornecimento de notas de aula e materiais, imagens e clips de vídeo;
- áreas do grupo de trabalho compartilhados - possibilita os usuários compartilharem arquivos e se comunicarem;
- suporte para estudantes - possibilita a comunicação entre os tutores e seus estudantes, fornecimentos de materiais didáticos e alguma forma de tirar as dúvidas dos alunos;
- gestão e acompanhamento dos estudantes - sistema de autenticação para permitir que apenas estudantes tenham acesso aos cursos;
- ferramentas para o estudante - por exemplo agendas e calendários eletrônicos;
- aparência consistente e personalizável - uma interface padrão de fácil utilização, permitindo personalização, mas com um modo de utilização básico.

A utilização de AVAs traz diversas vantagens como cita TAJRA (2001, p.153): acessibilidade a fontes inesgotáveis de assuntos para pesquisas, páginas educacionais específicas para a pesquisa escolar, comunicação e interação com outras escolas, estímulo para pesquisar a partir de temas previamente definidos ou a partir da curiosidade dos próprios alunos, estímulo ao raciocínio lógico, troca de experiências entre professores/professores, aluno/aluno e professor/aluno, dentre outras.

Carvalho (2013) afirma que AVAs integram funcionalidades de comunicação e partilha de informações e isso, permite aceder à aprendizagem de uma forma flexível; em qualquer espaço (*anywhere*) e em qualquer hora (*anytime*), o autor complementa que:

“Um AVA deve, por um lado, enfatizar a aprendizagem através da integração de ferramentas interativas e comunicativas, da partilha de conteúdos multimédia, do alojamento de trabalhos e projetos, da integração de ferramentas de aprendizagem colaborativa, e por outro, deve proporcionar estratégias que potenciem a participação ativa e significativa dos alunos, abranger possibilidades didáticas de aprendizagem individual e em grupo, criar novos acessos a websites como forma de enriquecer o conhecimento, possuir ferramentas de controlo de acesso e registro de utilizadores e de gestão de grupos de trabalho” (CARVALHO, 2013, p. 41).

Os AVAs, geralmente, utilizam diversas ferramentas para apoio ao ensino, como já foi citado anteriormente, tais como fóruns, chats, wikis, glossários, portfólios, enquetes, questionários, entre outros. Essas ferramentas, de acordo com Masetto (2012), são recursos em linguagem digital e podem colaborar significativamente para tornar a educação mais eficiente e eficaz.

2.3 Gamificação

Ao longo da história, buscamos metodologias inovadoras para auxiliar a educação. Uma das metodologias que mais diferem do tradicional método de ensino é a utilização de jogos para o ensino e aprendizagem.

Surgiu em 2002, por meio de Nick Pelling, programador de computadores e inventor britânico, o termo Gamificação. Fardo (2013) define a Gamificação como o emprego de conceitos e técnicas criadas e utilizadas em jogos para auxiliar na educação. Alguns exemplos são:

- Sistema de Pontos: são abertos, diretos e motivacionais, permitindo a utilização de vários tipos diferentes de pontuação, de acordo com o objetivo proposto.
- Medalhas e Conquistas: tratam-se de uma representação visual de alguma realização/conquista do usuário no sistema. Os usuários querem receber medalhas dentro de um ambiente por diversos motivos, para muitos, o objetivo é a experiência agradável de receber a medalha ou por “coleccionar” medalhas.
- Desafios e Missões: são os elementos que orientam os usuários sobre as atividades que devem ser realizadas dentro de um sistema. É importante que existam desafios para os usuários completarem, pois isso fará com que exista algo interessante para ele realizar enquanto interage com o sistema.
- Níveis: indicam o progresso do usuário dentro do sistema.
- Rankings: seu propósito principal é a comparação entre os jogadores/usuários envolvidos.
- Personalização: permite o usuário transformar e personalizar itens que compõem o sistema de acordo com seu gosto, promovendo motivação, engajamento, sentimento de posse e controle sobre o sistema.

Para Halliwell (2013), o objetivo da Gamificação não é transformar tudo em um jogo, mas sim, encontrar a diversão, encontrar os aspectos ‘jogáveis’ de um problema, quaisquer que sejam, e usá-los para criar um ambiente que mova as pessoas um pouco mais em direção a um objetivo que tenham criado. Um exemplo bem interessante do uso de gamificação é o Duolingo³ (AHN, 2013), uma plataforma online de aprendizagem em línguas que trabalha com o conceito de conhecimento coletivo e voluntário. No Duolingo, usuários podem ganhar pontos com as respostas corretas e lições completadas, assim como perder pontos a cada resposta incorreta. O mesmo ainda atribui status de reconhecimento de acordo o conhecimento obtido durante o jogo. Veja mais na subseção 2.4.3.

Um estudo realizado na Universidade da Cidade de Nova York afirmou que

³ O Duolingo é uma plataforma para ensino de idiomas gratuito, disponível em <www.duolingo.com>

2.4 Trabalhos Relacionados

Os Ambientes Virtuais de Aprendizagem vêm sendo utilizados na educação, principalmente como uma ferramenta mais dinâmica comparada as metodologias de ensino tradicionais e são de grande potencial na área da educação. Existem diversas pesquisas voltadas a aplicação de AVAs para apoiar o processo de ensino-aprendizagem e, em especial, a Matemática se destaca entre elas.

2.4.1 *The one world schoolhouse: Education reimagined*

Em um trabalho iniciado em 2006, Khan (2012) fundou a chamada KhanAcademy⁴, organização educacional que tem por objetivo oferecer exercícios, vídeos de instrução e um painel de aprendizado personalizado que habilita os estudantes a aprender no seu próprio ritmo dentro e fora da sala de aula. Em sua plataforma, são abordados assuntos como matemática, ciência, programação de computadores, história, história da arte, economia e muito mais (KHAN, 2012).

No ambiente, o desempenho do estudante é representado por medalhas. De acordo com o site da organização, medalhas e insígnias estimulam o aprendizado de maneira lúdica. As estatísticas mostram o quanto de trabalho o estudante está fazendo a cada dia, o quanto o estudante está focado em áreas de habilidades e tópicos e as habilidades que o estudante concluiu. Com os relatórios gerados pela plataforma, o tutor pode acompanhar todos os passos do aluno.

Durante uma conferência TED⁵ em 2011, denominada *Let's use video to reinvent education* (KHAN, 2011), Khan fala sobre o funcionamento da plataforma e também do provável motivo do seu sucesso. Segundo Khan (2011), o diferencial da plataforma está em dois pontos importantes, a aprendizagem auto-ritmada e nos dados fornecidos aos tutores sobre o aprendizado de seus alunos.

Em relação a aprendizagem auto-ritmada, Khan afirma que:

“Então quando fala-se em aprendizagem auto-ritmada, isso faz sentido para todo mundo – na chamada aprendizagem diferenciada – mas é meio maluco quando você vê isso na sala de aula. Porque toda vez que fizemos isso, em cada sala de aula que fizemos, repetidas vezes, depois de cinco dias nisso, há um grupo de garotos que está adiantado e há outro grupo de garotos um pouco atrás. E num modelo tradicional, se você fizesse uma avaliação pontual, diria: “Esses são garotos inteligentes, aqueles são garotos lerdos. Talvez eles devessem ser acompanhados de forma diferente. Talvez devêssemos coloca-los em salas diferentes.” Mas quando você deixa cada aluno trabalhar em seu próprio ritmo – e vemos isso repetidas vezes – você vê alunos que tomam um tempo extra em um

⁴ Plataforma de aprendizagem disponível em: <www.khanacademy.org>

⁵ É uma série de conferências realizadas na Europa, Ásia e Américas pela fundação Sapling com o objetivo de disseminar ideias que podem mudar o mundo.

conceito ou outro, mas uma vez que adquirem esse conceito, eles apenas vão adiante. E assim os mesmos garotos que você pensava que eram lerdos semanas atrás, agora você pensa que são inteligentes. E vemos isso repetidas vezes. E isso faz você se perguntar quantos estereótipos que talvez vários de nós recebemos eram apenas devido a uma coincidência de tempo” (KHAN, 2011, 13:29, Tradução Livre).

Khan (2011) também fala sobre a importância dos dados fornecidos pelos tutores:

“[...] Nosso paradigma é armar os professores com a maior quantidade de dados possível – dados que, em quase qualquer outro campo, são esperados, se você trabalha com finanças ou propaganda ou fabricação. E assim os professores podem realmente diagnosticar o que está errado com os alunos de maneira que podem fazer suas interações mais produtivas possível. Agora os professores sabem exatamente o que os alunos têm feito, quanto tempo eles gastam todo dia, a quais vídeos assistiram, quando pausaram os vídeos, o que fez que parassem de ver, quais exercícios estavam fazendo, no que eles estavam se focando? [...]” (KHAN, 2011, 12:26, Tradução Livre).

O trabalho de Khan (2012), assim como o apresentado nesta monografia, apresenta o uso de um AVA para auxiliar na educação. Dessa forma, esse trabalho servirá como referência para abordar o uso da aprendizagem auto-ritmada na educação matemática, assim como, para definir as informações que deverão ser apresentadas aos tutores sobre a evolução da aprendizagem de seus alunos na plataforma fruto deste trabalho. Contudo, os vídeos que fizeram da Khan Academy tão popular, não farão parte desse trabalho.

O aspecto mais importante a se considerar aqui está na forma como as duas plataformas lidam com Obstáculos Epistemológicos. Segundo Bachelard (1996), durante o ato do conhecimento, ocorrem “lentidões e conflitos” que levam o aluno a parar diante do problema. A esta “inércia” é que foi relacionado o conceito. Na metodologia criada por Khan (2012), quando a plataforma é aplicada dentro da sala de aula, o professor pode identificar através da ferramenta, os alunos que estão com esses obstáculos e o mesmo pode intervir para ajudar o aluno a superar essa barreira.

No trabalho apresentado aqui, essa barreira será superada quando o sistema, ao identificar o obstáculo enfrentado, indicar ao aluno a existência desse obstáculo e o(s) conteúdo(s) que ele possui deficiência (causador(es) da barreira) para ele assim poder pausar o conteúdo que está estudando e voltar ou não a estudar o(s) conteúdo(s) que o sistema indicar.

2.4.2 *ActiveMath: A generic and adaptive web-based learning environment*

O projeto ActiveMath visa apoiar a aprendizagem verdadeiramente interativa, exploratória, e assume que o estudante deve ser responsável por seu aprendizado, até

certo ponto. Portanto, uma relativa liberdade para navegar através de um curso e para as escolhas de aprendizagem lhes é dada e, por padrão, o modelo de usuário é inspecionável e modificável (MELIS et al., 2001).

Melis et al. (2001) afirmam que a maioria dos sistemas tutores inteligentes não contam com uma escolha de adaptação de conteúdos e isso, segundo ele, pode influenciar quando o público-alvo for alunos de faculdades e universidade, já que, diferentemente das escolas de ensino fundamental, um mesmo assunto é ensinado de forma diferente para diferentes grupos de utilizadores e em contextos diferentes (MELIS et al., 2001).

Para conseguir um ambiente dinâmico de aprendizagem, ActiveMath utiliza regras pedagógicas que definem em quais momentos determinadas funcionalidades do sistema estarão disponíveis, em que ordem os conteúdos serão apresentados para os alunos e como os mesmos deverão ser apresentados. O trabalho referido, assim como o desenvolvido por Khan (2012) descrito anteriormente, descreve a criação de um AVA. Mas também, o mesmo utiliza técnicas que permitem a geração dinâmica de cursos para os alunos.

O trabalho desenvolvido por Melis et al. (2001) assim como o descrito aqui, apresenta um subsistema de exercícios que suporta diagnósticos de erros e equívocos dos estudantes, que gera estratégias tutoriais configuráveis para o *feedback*.

2.4.3 *Duolingo: Learn a Language for Free while Helping to Translate the Web*

Nesse trabalho desenvolvido por Ahn (2013), é apresentado o Duolingo, uma plataforma de ensino de idiomas e tradução automática de documentos. O ambiente funciona de maneira que os usuários progridam nas lições ao mesmo tempo que traduzem conteúdo real da internet.

O método utilizado pela plataforma se caracteriza pela lições fragmentadas, pelas quais os alunos, através do método de repetição, fixam o conteúdo da língua estudada. A medida que o usuário avança, ele progride em uma árvore de habilidades que leva o estudante gradativamente ao fim do curso, enquanto oferece a opção de voltar atrás para refazer lições antigas que já poderiam ter sido esquecidas.

Um estudo realizado na Universidade da Cidade de Nova York (VESSELINOV; GREGO, 2012) disse que 34 horas gastas no Duolingo igualou-se a um semestre de um curso de línguas.

Uma das características mais marcantes dessa plataforma é quantidade de técnicas de Gamificação empregadas. Possui sistema de pontuação, níveis, rankings, missões, medalhas, personalização, entre outras.

Assim como a plataforma desenvolvida por Ahn (2013), o ambiente que desenvol-

veremos também contará com técnicas de Gamificação que serão utilizadas para melhorar a experiência do usuário, assim como um para motivar os usuários durante sua utilização.

2.4.4 *Resolução de problemas em ambientes virtuais de aprendizagem num curso de licenciatura em matemática na modalidade a distância*

O trabalho desenvolvido por Dutra (2011), trata da utilização da metodologia de Resolução de Problemas (apresentada na subseção 2.1.2) em um AVA com o objetivo de investigar as contribuições que essa junção pode trazer para um curso de Licenciatura em Matemática da Universidade Federal de Ouro Preto (UFOP), na Educação à Distância. Para isso, o ambiente desenvolvido por Dutra (2011) utiliza de fóruns semanais para discussão e resolução dos problemas, além de chats, utilizados ao final de algumas atividades, e um questionário final a ser respondido pelos alunos na última semana de aula.

Essa metodologia, segundo Dutra (2011), funciona através dos seguintes passos:

1. A atividade é postada na Plataforma Moodle⁶ no início da semana, pela manhã (segunda-feira). Assim, as atividades são distribuídas aos alunos para que possam ler, interpretar e entender o problema.
2. Os alunos passam a semana postando suas resoluções dos problemas e discutindo-as no fórum com os colegas, por meio da Plataforma Moodle.
3. A pesquisadora observa, incentivava e participava do processo de discussão. Ajuda nos problemas secundários, dando *feedback* das resoluções postadas, respondendo e fazendo perguntas, tirando dúvidas, acompanhando de perto as discussões entre os alunos no fórum.
4. As impressões dos alunos sobre os problemas, no início da semana seguinte, e a formalização dos resultados são apresentadas nos chats semanais. Trata-se de uma plenária virtual para discutir os problemas, finalizando-a com uma solução aceita por todos.
5. Uma resolução, após o chat, é postada na Plataforma Moodle, para todos os pesquisados, observando os conteúdos apresentados nos problemas.

O ambiente desenvolvido por Dutra (2011) inspirou este trabalho na utilização da metodologia de Resolução de Problema aplicada num AVA através de fóruns e chats,

⁶ “A palavra Moodle referia-se originalmente ao acrônimo: ‘Modular Object-Oriented Dynamic Learning Environment’(...). Em inglês a palavra Moodle é também um verbo que descreve a ação que, com frequência, conduz a resultados criativos, de deambular com preguiça, enquanto se faz com gosto o que for aparecendo para fazer”. O Moodle deu o nome a uma plataforma de e-learning, de utilização livre e código fonte aberto, pela mão de Martin Dougiamas (ALVES et al., 2009).

entretanto, a forma como essas ferramentas darão suporte ao ensino em nossa metodologia será adaptada. O fórum que na metodologia de Dutra (2011) era utilizado para a postagem da resolução dos problemas, será utilizado como ferramenta para os alunos tirarem dúvidas sobre os problemas apresentados, já o chat utilizado para discutir os problemas na metodologia de Dutra (2011), será utilizado para permitir que alunos que concluíram determinado conteúdo com certa proficiência, possa ajudar outros alunos que enfrentem obstáculos epistemológicos ao longo do aprendizado desse conteúdo.

2.5 Considerações finais do capítulo

O sistema que propomos desenvolver faz uso da metodologia de resolução de problemas, possibilitando o aluno adquirir novos conhecimentos além daqueles estudados no momento. Por se tratar de um AVA, o mesmo possibilitará ao aluno uma maior interação com o professor e outros alunos, além da possibilidade de uma aprendizagem auto-ritmada. A aplicação da Gamificação no sistema, servirá para desenvolver engajamento, participação, e comprometimento entre os usuários do sistema.

3 PROCEDIMENTOS METODOLÓGICOS

Os procedimentos metodológicos definem as principais etapas realizadas para o desenvolvimento deste trabalho, incluindo as pesquisas, o desenvolvimento e avaliação do *software*. Nas seções a seguir, descrevemos essas etapas.

3.1 Definição do Processo

Processos de *software* são utilizados pelos engenheiros de *software* para controlar e coordenar projetos de desenvolvimento de *softwares* reais (TALMA, 2006). Pádua (2003) descreve um processo como um conjunto de passos parcialmente ordenados, constituídos por atividades, métodos, práticas e transformações, usados para atingir uma meta. Desta forma, um modelo de processo de *software* é uma descrição simplificada de um processo, sendo também uma representação abstrata do mesmo para explicar as diferentes abordagens de desenvolvimento (SOMMERVILLE, 2003).

Processos de software são complexos e dependem do julgamento humano como em qualquer processo intelectual. Sendo assim, não existe um processo de software ideal, todos são desenvolvidos de maneiras diferentes por cada organização (SOMMERVILLE, 2003).

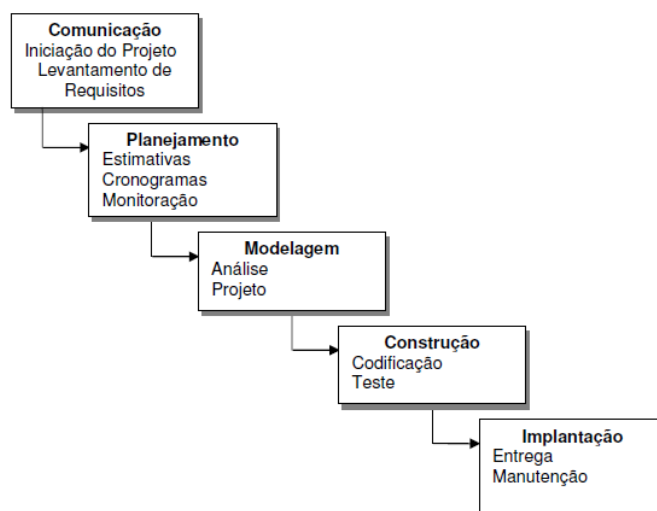
O processo de *software* utilizado para desenvolver o sistema foi baseado no modelo cascata, também chamado de ciclo de vida clássico, proposto por Royce em 1970. Neste modelo, as fases são sistematicamente seguidas de maneira sequencial (PRESSMAN, 2006).

O modelo inicia com a fase de especificação de requisitos, passando pelo planejamento, modelagem, construção e implantação, finalizando na manutenção progressiva do software, como apresentamos na Modelo Cascata.

As vantagens desse modelo se devem ao fato de que só se avança para a tarefa seguinte quando o cliente valida e aceita os produtos finais da tarefa atual, facilitando assim a compreensão adquirida ao longo do projeto, além de facilitar o processo de criação da documentação para o sistema (PRESSMAN, 2006). Já as principais desvantagens, segundo Pressman (2006), se devem ao fato de que os projetos reais raramente seguem o fluxo sequencial ao qual o modelo propõe. Este modelo exige ainda que todos os requisitos sejam estabelecidos na fase inicial, fato que geralmente é difícil tanto para o cliente quanto para o desenvolvedor, pois os requisitos geralmente estão em constante mudança. Outro grande problema com esse modelo é que o cliente só recebe uma versão executável do sistema no final de todo o processo de desenvolvimento, o que não agrada a muitos clientes.

Levando em consideração as vantagens e desvantagens antes citadas, esse modelo foi escolhido como base para o processo por facilitar o desenvolvimento de uma documentação mais detalhada e principalmente pela equipe de desenvolvimento ser formada por uma única pessoa, o autor deste trabalho, impossibilitando a divisão de tarefas característica de metodologias ágeis ⁷.

Figura 1 – Modelo Cascata



Fonte: Pressman (2006)

Dividimos o processo em nove atividades. A seguir, descreveremos brevemente cada uma delas:

⁷ Metodologias de desenvolvimento de software que tem enfoque nas pessoas e não em processos ou algoritmos, além de uma preocupação menor em documentação e maior em implementação (SOARES, 2004).

1. Iniciar Projeto: nessa atividade identificamos os objetivos do projeto, suas restrições, envolvidos, a organização da equipe de desenvolvimento e por último realizamos uma reunião de início do projeto.
2. Requisitos: nessa atividade realizamos o levantamento e análise dos requisitos, sua documentação, verificação e validação, além de um estudo para saber através dos requisitos se o projeto é viável.
3. Projeto: nessa atividade desenvolvemos o projeto da arquitetura do sistema, o planejamento dos módulos, o projeto da interface gráfica com o usuário e como serão persistidos as entidades do sistema.
4. Implementação: nessa atividade escolhemos primeiramente um módulo do sistema para implementar, em seguida planejamos como ocorrerá a implementação desse módulo, implementamos o módulo, e por último realizamos a integração de módulo ao resto do sistema. Esse ciclo é repetido até que todos os módulos estejam desenvolvidos.
5. Testes: durante essa atividade realizaremos o planejamento dos testes para o sistema, em seguida executaremos os testes para cada módulo do sistema e para o sistema como um todo, e por último iremos gerar um relatório com os resultados dos testes.
6. Implantação: após o sistema desenvolvido e devidamente testado, ele segue para a implantação, em que prepararemos o ambiente onde o sistema será executado e realizaremos a implantação. Após a implantação, realizaremos testes de aceitação e a criação de um manual para o sistema.
7. Gerenciamento do Projeto: essa atividade ocorre durante todo o desenvolvimento do sistema e tem por objetivo permitir que o projeto consiga ser concluído dentro do prazo e da qualidade desejada.
8. Avaliação do Processo: essa atividade é executada durante todo o desenvolvimento do projeto e permite que o processo utilizado seja constantemente avaliado e adaptado conforme segue o desenvolvimento do projeto.
9. Encerrar Projeto: nessa última atividade será realizado apenas uma reunião em que será oficializado o final do projeto.

O processo desenvolvido está disponível em [<www.askmath.quixada.ufc.br/static/process/>](http://www.askmath.quixada.ufc.br/static/process/).

3.2 Levantamento e Análise de Requisitos

O Levantamento e Análise de Requisitos é a fase do desenvolvimento de um software em que o analista verifica junto ao usuário, quais as necessidades, condições e princípios que

o *software* deverá atender (MATUDA; BEGOSSO, 2013). Essa fase possibilitou conhecer e estudar as necessidades do cliente, assim como as restrições que o software estará sujeito.

Para realizar a coleta dos requisitos, optamos por utilizar entrevistas com o cliente. Nessas entrevistas, que se caracterizarão como semi-estruturadas, já que foram guiadas por um roteiro previamente elaborado, composto por questões abertas (BELEI et al., 2008), foi possível obter os requisitos do sistema, assim como o público-alvo a quem o sistema atenderá. Essa técnica foi utilizada porque permitia uma organização flexível e ampliação dos questionamentos à medida que as informações foram sendo fornecidas pelo cliente (FUJISAWA, 2000).

O ambiente atenderá quatro tipos de usuários que formaram seu público-alvo, serão administradores, professores, assistentes, e estudantes. Os administradores serão responsáveis por cadastrar novos professores, assim como realizar a manutenção diária do sistema. Os professores serão responsáveis por definir as disciplinas e lições que farão parte do sistema. Após as disciplinas e lições definidas pelos professores, os assistentes serão responsáveis por manter os problemas de cada lição. Os estudantes resolverão os problemas desenvolvidos pelos assistentes e poderão postar dúvidas sobre qualquer conteúdo no fórum de discussões. Quando o ambiente for aplicado em sala de aula, os professores serão ainda responsáveis por gerenciar suas turmas e acompanhar o andamento do progresso de cada um de seus estudantes, já os assistentes serão responsáveis por tirar dúvidas dos estudantes no fórum de discussões.

Para uma melhor compreensão do público-alvo, foram criadas Personas (PRUITT; GRUDIN, 2003), personagens fictícios usados para caracterizar os papéis dos diferentes usuários do sistema (GUERRA, 2010). Cada persona criada possui um nome, hábitos, histórias pessoais, motivações, objetivos, entre outras (ver APÊNDICE B - DOCUMENTO DE PERSONAS). A escolha dessa técnica deu-se pelo fato de que ela permitia ao desenvolvedor saber mais precisamente para quem ele deveria construir o sistema, além de permitir uma distinção maior do público-alvo e, dessa forma, aprofundar-se nos interesses individuais de cada um.

Apresentaremos a seguir, os principais requisitos levantados durante essa etapa:

- Hierarquia de Conteúdos: no sistema, deverão existir disciplinas, cada disciplina deverá possuir lições e cada lição será formada por problemas. Os professores poderão cadastrar várias disciplinas e lições, já os assistentes ficarão responsáveis por adicionar os problemas nas lições. Quando o estudante entrar no sistema, ele poderá ver somente as lições de uma disciplina por vez, podendo alternar entre disciplinas. Para um estudante começar a responder os problemas, ele deverá escolher inicialmente a disciplina e em seguida a lição.
- Estrutura do Problema: cada problema possuirá uma descrição e vários itens.

Os problemas serão somente de múltipla escolha e poderão possuir no mínimo dois e no máximo cinco itens, essa quantidade ficará a critério do assistente que realizará a adição do problema no sistema.

- Saltar Problemas: o sistema deverá permitir que o estudante possa saltar os problemas e rever os saltos realizados, isso com algumas restrições em relação a quantidade de vezes.
- Pedir Ajuda: para cada problema, o estudante poderá possuir uma ajuda. No momento em que o assistente estiver adicionando o problema, ele poderá ou não adicionar um texto de ajuda para o estudante, isso fica a critério dele. Quando o estudante estiver resolvendo os problemas, ele terá um botão que quando acionado, mostrará o texto de ajuda. O sistema deverá salvar a quantidade de vezes que o estudante pedir ajuda e em quais problemas.
- Deficiências: é possível, a partir de um item respondido incorretamente em um problema, identificar a deficiência do estudante, por isso é fundamental que no momento que o assistente estiver adicionando os itens do problema, ele possa escolher para cada item incorreto, uma ou várias lições, essas lições ficarão sendo as supostas deficiências do aluno caso ele responda incorretamente aquele problema marcando aquele determinado item.
- Fórum: caso o aluno possua dúvidas em relação aos conteúdos, o sistema deverá possuir um fórum de discussões em que esse aluno possa postar essas dúvidas, para que professores, assistentes ou outros estudantes possam lhes ajudar.

Após o levantamento dos requisitos, foi realizada a análise dos mesmos. Nessa análise, os requisitos foram agrupados em categorias. As categorias utilizadas são descritas por Sommerville (2003) como:

- Requisitos Funcionais – especificam ações que um sistema deve ser capaz de executar, sem levar em consideração restrições físicas. Os requisitos funcionais especificam, portanto, o comportamento de entrada e saída de um sistema.
- Requisitos Não Funcionais – descrevem apenas atributos do sistema ou atributos do ambiente do sistema, como segurança, desempenho, usabilidade e confiabilidade.
- Requisitos de Domínio – são os requisitos do domínio da aplicação do sistema e que refletem características desse domínio.

Após esse agrupamento, os requisitos funcionais foram representados em Casos de Uso (JACOBSON, 1992). Um caso de uso identifica os agentes envolvidos em uma interação e especifica o tipo de interação utilizando anotações sugeridas pela Unified Modeling Language (UML) Sommerville (2003). Em seguida, foi realizada a documentação dos requisitos (ver APÊNDICE A - DOCUMENTO DE REQUISITOS).

No final dessa etapa, ocorreu a Validação dos Requisitos junto ao cliente. A Validação dos Requisitos é definida como o processo que certifica que o modelo de requisitos gerado esteja consistente com as necessidades e intenções de clientes e usuários (PAIM, 2003). Esta etapa permitiu que os requisitos coletados e documentados estivesse de acordo com o que o cliente solicitou.

3.3 Projeto do Sistema

O Projeto de Software é à atividade de engenharia cujo foco é definir “como” os requisitos estabelecidos do projeto devem ser implementados no software (PRESSMAN, 2006). O objetivo da atividade de projetar é gerar um modelo ou representação que apresente solidez, comodidade e deleite (PRESSMAN, 2006).

Nesta fase, definimos como será aplicado o conhecimento obtido na pesquisa bibliográfica para se desenvolver o sistema. Para isto, definimos a arquitetura de software e as ferramentas que serão utilizadas durante o desenvolvimento do sistema. Nas seções a seguir, descrevemos um pouco sob cada um.

3.3.1 Arquitetura

Por arquitetura de software, entende-se a estrutura ou a organização de componentes de módulos, a maneira através da qual esses componentes interagem e a estrutura de dados que será usada pelos componentes (PRESSMAN, 2006).

A arquitetura utilizada baseia-se na arquitetura Cliente-Servidor (DAVID et al., 2013), em que o processamento é dividido em processos distintos. Um processo é responsável pela manutenção da informação (servidor) e os outros são responsáveis pela captação de dados (clientes). Nessa arquitetura, os clientes enviam pedidos para o servidor, e este por sua vez processa estes dados e envia as respostas dos pedidos aos clientes.

Este modelo de arquitetura facilitará na manutenção do sistema, tendo em vista que toda atualização só necessitará ser realizada no servidor e automaticamente a mesma se propagará para todos os clientes. Com todos os recursos centralizados no servidor, podemos também ter um maior ganho com segurança, já que podemos centralizar os nossos esforços para manter a segurança das informações em apenas um único ponto, além de possibilitar que apenas cliente credenciados possam acessar e/ou alterar essas informações. Uma das outras grandes vantagens que temos ao utilizar esse modelo, é que a medida que a quantidade de clientes aumente, será possível suprir esses clientes sem necessitar realizar nenhuma modificação essencial.

Para uma visualização mais detalhada da arquitetura de software definida, visitar o APÊNDICE C - DOCUMENTO DE ARQUITETURA.

3.3.2 Ferramentas

A análise do sistema foi feita com o auxílio da ferramenta de criação de diagramas Astah (ASTAH, 2016), a implementação com a linguagem Python (VANROSSUM; DRAKE, 2010), com o sistema de gerenciamento de banco de dados PostgreSQL (MOMJIAN, 2001) e a camada de aplicação através da utilização do framework Django (DJANGO, 2016). Assim como a utilização do módulo Rosseta (ROSETTA, 2016) para permitir a internacionalização do sistema.

A seguir, apresentaremos a lista das ferramentas e das tecnologias utilizadas para o desenvolvimento do projeto:

- a) Astah: para a modelagem baseada em UML (Unified Modeling Language) do sistema.
- b) Python: linguagem de programação para implementação do sistema.
- c) Django: framework web responsável pela camada de aplicação.
- d) Rosetta: aplicação desenvolvida em Django que facilitará a tradução do projeto para diversas línguas.
- e) PostgreSQL: como banco de dados para armazenamento e consulta de informações.
- f) Metro UI CSS: framework que faz uso de HTML, Cascading Style Sheet (CSS) e Javascript para criação do front-end do sistema.
- g) MathJax: é uma engine⁸ de código aberto desenvolvido em javascript na forma de um plugin para incluir equações matemáticas em todos os navegadores, esse plugin aceita expressões em MathML e Latex.

Essas ferramentas foram selecionadas por se tratarem, algumas, de ferramentas Open Source, ou seja, que seu código-fonte pode ser alterado para diferentes fins, possibilitando assim que qualquer um consulte, examine ou as modifique, e outras por serem ferramentas que possibilitam um rápido desenvolvimento.

No final dessa etapa, foi gerado o Plano de Projeto, esse documento guiará o desenvolvedor durante todo o processo de desenvolvimento.

3.4 Implementação do Sistema

A implementação envolve as atividades de codificação, compilação e integração. A codificação visa traduzir o design num programa, utilizando linguagens e ferramentas adequadas. A codificação deve refletir a estrutura e o comportamento descrito no projeto.

⁸ Uma biblioteca ou pacote de funcionalidades que são utilizadas para facilitar o desenvolvimento de alguma tecnologia.

Os componentes arquiteturais devem ser codificados de forma independente e depois integrados (AGUIAR, 2012).

3.5 Verificação e Validação

Essa etapa destina-se a mostrar que o sistema está de acordo com a especificação e que ele atende às expectativas de clientes e usuários, além de assegurar que o programa está fazendo aquilo que foi definido na sua especificação e que não possui erros de execução (AGUIAR, 2012).

Durante essa fase, serão realizados Testes de Unidade e Integração em cada modulo do sistema, assim como Testes de Sistema no sistema já integrado.

Aniche (2014) define essas categorias de Testes de Software como:

- Teste de Unidade – é aquele que testa uma única unidade do sistema. Ele a testa de maneira isolada, geralmente simulando as prováveis dependências que aquela unidade tem. Em sistemas orientados a objetos, é comum que a unidade seja uma classe. Ou seja, quando queremos escrever testes de unidade para a classe Pedido, essa bateria de testes testará o funcionamento da classe Pedido, isolada, sem interações com outras classes.
- Teste de Integração – é aquele que testa a integração entre duas partes do seu sistema. Os testes que você escreve para a sua classe PedidoDao, por exemplo, em que seu teste vai até o banco de dados, é um teste de integração. Afinal, você está testando a integração do seu sistema com o sistema externo, que é o banco de dados. Testes que garantem que suas classes comunicam-se bem com serviços web, escrevem arquivos texto, ou mesmo mandam mensagens via socket são considerados testes de integração.
- Teste de Sistema – é aquele que garante que o sistema funciona como um todo. Este nível de teste está interessado se o sistema funciona como um todo, com todas as unidades trabalhando juntas. Ele é comumente chamado de teste de caixa preta, já que durante o teste não se importa com a forma que os dados são processados, mas sim se as saídas estão de acordo com o que era esperado para entradas.

3.6 Definição dos Conteúdo para o Sistema

Após o sistema estar verificado e validado, ele terá que possuir conteúdos para ser utilizado pelo usuário final durante a fase de aplicação. A aplicação da primeira versão do sistema está planejada para ocorrer com uma turma de matemática da Universidade Federal

do Ceará - Campus Quixadá. Decidimos optar por deixar os monitores⁹ da disciplina desenvolverem os conteúdos que serão utilizados no sistema durante essa fase.

Os monitores passarão por um treinamento, ao qual aprenderão a utilizar o sistema para, assim, adicionar os conteúdos desenvolvidos.

3.7 Aplicação da Solução na Universidade Federal do Ceará - Campus Quixadá

Aplicaremos o ambiente desenvolvido em duas turmas de Matemática Básica da UFC - Campus Quixadá. Nessas turmas, que está previsto que sejam formadas por aproximadamente trinta estudantes cada, dividiremos os estudantes de cada turma em dois grupos, Grupo A1 e Grupo B1 para a primeira turma e Grupo A2 e B2 para a segunda turma. Os Grupos A1 e A2, serão formados pela metade dos estudantes de cada turma, escolhidos aleatoriamente, e os Grupos B1 e B2 pelos estudantes restantes. Esta divisão, deve-se ao fato de que ao final da experiência de utilização do ambiente pelos estudantes, realizaremos uma comparação no desempenho entre estudantes que utilizaram o sistema com os que não utilizaram. Dessa forma, o sistema será utilizado pelos estudantes dos Grupos A1 e A2, enquanto que os Grupos B1 e B2 (grupos de controle) continuaram estudando da mesma forma.

O objetivo dessa experiência é analisar o impacto que o uso do ambiente desenvolvido tem na aprendizagem dos estudantes. Os dados que iremos utilizar para atingirmos os objetivos com essa experiência serão providos de duas avaliações realizadas por todos os alunos das duas turmas antes e depois a utilização do ambiente pelos Grupos A1 e A2.

3.8 Cronograma de Execução

Tabela 1 – Cronograma de Execução

ATIVIDADES	2015						2016								
	Jan	Fev	Mar	Abr	Mai	Jun - Dez	Jan - Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Definição do Processo	x														
Levantamento e Análise dos Requisitos	x	x	x												
Projeto do Sistema				x	x										
Implementação do Sistema						x	x	x	x	x					
Verificação e Validação											x				
Desenvolver o conteúdo para o sistema								x	x	x	x	x			
Aplicação na UFC - Campus Quixadá												x	x	x	
Definição do Projeto de Pesquisa								x							
Defesa do Projeto de Pesquisa										x					
Ajustes Solicitados											x				
Análise dos Resultados Obtidos na Aplicação.															x
Defesa do Trabalho Final															x

⁹ É o aluno de graduação concursado para exercer, juntamente com o professor, atividades técnico-didáticas condizentes com o seu grau de conhecimento junto à determinada disciplina, já por ele cursada.

4 RESULTADOS PRELIMINARES

Nesta seção apresentaremos o andamento deste trabalho até o presente momento.

Até o momento, de concluído, temos o processo que está sendo utilizado durante o desenvolvimento do sistema, os requisitos do sistema que já foram coletados, analisados e validados, além do próprio projeto do sistema.

Na fase que está em andamento, que é a fase de implementação, já temos os seguintes módulos concluídos:

- Gerenciador de Usuários: módulo responsável por gerenciar os usuários do sistema como professores, assistentes e alunos.
- Gerenciador de Turmas: módulo responsável por gerenciar as turmas de alunos do sistema.
- Gerenciador de Disciplinas: módulo responsável por gerenciar as disciplinas que serão cadastradas no sistema.
- Gerenciador de Lições: módulo responsável por gerenciar as lições que os professores irão cadastrar no sistema.
- Gerenciador de Questões: módulo responsável por gerenciar os problemas que os assistentes e professores poderão cadastrar para cada lição.
- Gerenciador de Pontuação: módulo responsável por gerenciar a pontuação ganha pelos alunos, assim como seu nível de experiência ao longo da utilização do sistema.
- Fórum: módulo responsável por permitir que alunos postem dúvidas dos mais variados assuntos relacionadas ao sistema, seja dúvidas em relação ao conteúdo apresentado em sala de aula, assim como informações sobre o sistema e sugestões.

Os módulos que ainda restam para serem desenvolvidos nesta fase são:

- Gerenciador de Progresso: módulo responsável por acompanhar o andamento de cada estudante durante seu aprendizado e identificar os obstáculos epistemológicos enfrentados, indicando ao aluno a existência desses obstáculos e o(s) conteúdo(s) que ele possui deficiência (causador(es) do obstáculo) para ele assim poder pausar o conteúdo que está estudando e voltar ou não a estudar o(s) conteúdo(s) que o sistema indicar.
- Gerador de Estatísticas: módulo responsável por gerar as estatísticas que o professor utilizará para acompanhar o andamento de suas turmas e alunos, assim como para o uso pelo estudante, que utilizará para acompanhar seu próprio progresso durante sua aprendizagem no sistema.

- Ranking: módulo responsável por manter um ranking¹⁰ com os posicionamentos dos alunos de acordo com seu desempenho no sistema durante a semana.

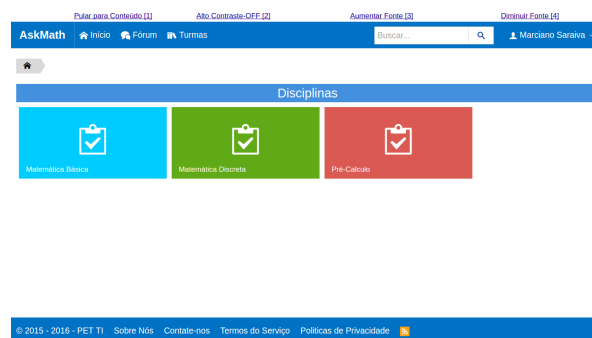
A seguir, apresentaremos algumas telas do sistema:

Figura 2 – Tela Inicial



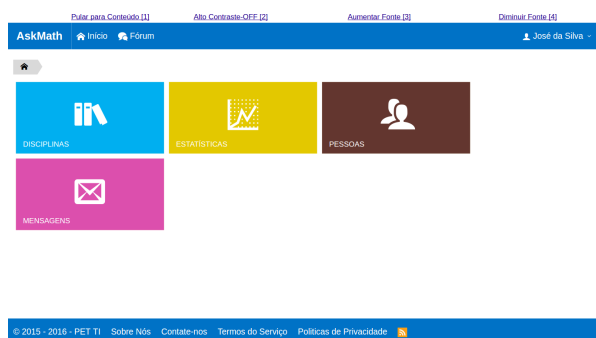
Fonte: <www.askmath.quixada.ufc.br>.

Figura 3 – Tela Inicial do Estudante



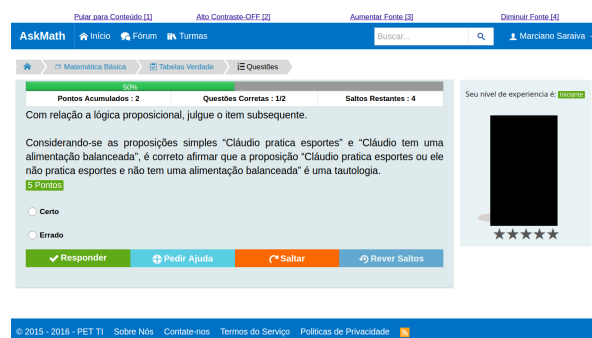
Fonte: <www.askmath.quixada.ufc.br>.

Figura 4 – Tela de Administração



Fonte: <www.askmath.quixada.ufc.br>.

Figura 5 – Tela de Problemas do Estudante



Fonte: <www.askmath.quixada.ufc.br>.

Os conteúdos que serão utilizados para popular o sistema, durante sua aplicação na UFC-Campus Quixadá, já estão sendo desenvolvidos pelos monitores citados anteriormente. Após a validação e verificação do sistema, esses conteúdos serão adicionados por esses monitores, que passaram por um treinamento para aprenderem a utilizar o sistema.

Referências

AGUIAR, A. P. H. d. Análise de requisitos para um consultório odontológico. MANOEL RIBAS/PR, 2012.

AHN, L. V. Duolingo: learn a language for free while helping to translate the web. In: ACM. **Proceedings of the 2013 international conference on Intelligent user interfaces**. [S.l.], 2013. p. 1–2.

¹⁰ É uma classificação ordenada de acordo com critérios determinados.

ALENCAR, S. de. **A gerência da criatividade: abrindo as janelas para a criatividade pessoal e nas organizações**. [S.l.]: Makron Books, 1996.

ALMEIDA, M. E. B. de. **ProInfo: Informática e formação de professores**. [S.l.]: Ministério da Educação, Secretaria de Educação a Distância, 2000.

ALVES, L.; BARROS, D.; OKADA, A. (Ed.). **Moodle: Estratégias Pedagógicas e Estudos de Casos**. Salvador, Brazil: EDUNEB, 2009. Published in Portuguese. Disponível em: <<http://oro.open.ac.uk/29585/>>.

ANICHE, M. **Unidade, integração ou sistema? Qual teste fazer?** 2014. Disponível em: <www.blog.caelum.com.br/unidade-integracao-ou-sistema-qual-teste-fazer/>. Acesso em: 26-05-2016.

ASTAH. **Astah Community**. 2016. Disponível em: <www.astah.change-vision.com>. Acesso em: 06-06-2016.

BACHELARD, G. A formação do espírito científico. **Rio de Janeiro: Contraponto**, p. 7–37, 1996.

BASSANEZI, R. C. **Ensino-aprendizagem com modelagem matemática: uma nova estratégia**. [S.l.]: Editora Contexto, 2002.

BELEI, R. A.; GIMENIZ-PASCHOAL, S. R.; NASCIMENTO, E. N.; MATSUMONO, P. H. V. R. O uso de entrevista, observação e videogravação em pesquisa qualitativa. **Cadernos de educação**, n. 30, 2008.

BITNER, N.; BITNER, J. Integrating technology into the classroom: Eight keys to success. **Journal of technology and teacher education**, v. 10, n. 1, p. 95–100, 2002.

BITZER, D.; BRAUNFELD, P.; LICHTENBERGER, W. Plato: An automatic teaching device. **Education, IRE Transactions on**, IEEE, v. 4, n. 4, p. 157–161, 1961.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E.; YERGEAU, F. Extensible markup language (xml). **World Wide Web Consortium Recommendation REC-xml-19980210**. <http://www.w3.org/TR/1998/REC-xml-19980210>, v. 16, p. 16, 1998.

BURAK, D. Modelagem matemática e a sala de aula. **Encontro Paranaense de Modelagem em Educação Matemática**, v. 1, p. 1–10, 2004.

CARVALHO, L. T. **Ambiente Virtual de Aprendizagem Matemática em contexto educativo**. Tese (Doutorado), 2013.

DANTE, L. R. Didática da resolução de problemas de matemática. **São Paulo: Ática**, v. 1, 1991.

DAVID, T.; GUERRAOUI, R.; TRIGONAKIS, V. Everything you always wanted to know about synchronization but were afraid to ask. In: ACM. **Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles**. [S.l.], 2013. p. 33–48.

DJANGO. **Official Django Site**. 2016. Disponível em: <www.djangoproject.com>. Acesso em: 06-06-2016.

DUTRA, D. S. d. A. Resolução de problemas em ambientes virtuais de aprendizagem num curso de licenciatura em matemática na modalidade a distância. Programa de Pós-Graduação em Educação Matemática. Departamento de Matemática, Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto., 2011.

FARDO, M. L. A gamificação aplicada em ambientes de aprendizagem. **RENOTE**, v. 11, n. 1, 2013.

FOSSA, J.; MENDES, I. Tendências atuais na educação matemática: experiências e perspectivas. **XIII Encontro de Pesquisa Educacional do Nordeste—Coleção EPEN**, v. 19, 1998.

FREIRE, P.; GUIMARÃES, S. Sobre educação (diálogos). Paz e Terra., 1982.

FUJISAWA, D. S. **Utilização de jogos e brincadeiras como recurso no atendimento fisioterapêutico de criança: implicações na formação do fisioterapeuta. 2000.** Tese (Doutorado) — Dissertação (Mestrado em Educação)-Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, Marília, 2000.

GUERRA, P. L. Colaboração para captura de requisitos: uma experiência com uso de personas. UFMG, 2010.

HALLIWELL, J. **Gamification for the clueless: how you can make the mundane more exciting by using Points, Badges and Leaderboards, and what this can do for your business.** [S.l.]: Amazon Digital Services, Inc., 2013. 39 p.

HAMMES, L. **TENDÊNCIAS METODOLÓGICAS ATUAIS NA PRÁTICA DE ENSINO.** Tese (Doutorado) — Universidade Federal de Santa Catarina, 2003.

JACOBSON, I. **Object-Oriented Software Engineering.** [S.l.]: São Paulo: Addison Wesley, 1992.

KHAN, S. **Let's use video to reinvent education.** 2011. Disponível em: <www.ted.com/talks/salman_khan_let_s_use_video_to_reinvent_education>. Acesso em: 09-05-2016.

KHAN, S. **The one world schoolhouse: Education reimaged.** [S.l.]: Twelve, 2012.

LOPES, A. O.; VEIGA, I. P. A. Aula expositiva: superando o tradicional. **Técnicas de ensino: por que não**, v. 2, p. 35–113, 1995.

MASETTO, M. T. **Competência pedagógica do professor universitário.** [S.l.]: Summus Editorial, 2012.

MATUDA, D. M.; BEGOSSO, L. C. Mapas mentais na engenharia de requisitos. **Anais SULCOMP**, 2013.

MELIS, E.; ANDRES, E.; BUDENBENDER, J.; FRISCHAUF, A.; GODUADZE, G.; LIBBRECHT, P.; POLLET, M.; ULLRICH, C. Activemath: A generic and adaptive web-based learning environment. **International Journal of Artificial Intelligence in Education (IJAIED)**, v. 12, p. 385–407, 2001.

MELIS, E.; SIEKMANN, J. Activemath: An intelligent tutoring system for mathematics. In: SPRINGER. **International Conference on Artificial Intelligence and Soft Computing.** [S.l.], 2004. p. 91–101.

- MELO, M. **Princípios e normas para a matemática escolar**. Tradução M. Rebelo. [S.l.]: Associação de Professores de Matemática, 2007. 466 p.
- MOMJIAN, B. **PostgreSQL: introduction and concepts**. [S.l.]: Addison-Wesley New York, 2001.
- OECD. Annual report of the Programme for International Student Assessment, **Low-Performing Students: Why They Fall Behind and How to Help Them Succeed**. 2016.
- OLEARY, R.; RAMSDEN, A. Virtual learning environments. **Learning and Teaching Support Network Generic Centre/ALT Guides, LTSN**. Retrieved July, v. 12, p. 2005, 2002.
- PÁDUA, W. d. Engenharia de software: fundamentos, métodos e padrões. **Rio de Janeiro: LTC**, 2003.
- PAIM, F. R. S. Uma metodologia para definição de requisitos em sistemas data warehouse. Universidade Federal de Pernambuco, 2003.
- PONTE, J. P. da. Ensino da matemática na sociedade de informação. 1997.
- PRESSMAN, R. S. **Engenharia de Software**. [S.l.]: São Paulo: Mcgraw-hill, 2006. 720 p.
- PRUITT, J.; GRUDIN, J. Personas: practice and theory. In: ACM. **Proceedings of the 2003 conference on Designing for user experiences**. [S.l.], 2003. p. 1–15.
- ROSETTA. **Rosetta Documentation**. 2016. Disponível em: <www.django-rosetta.readthedocs.io>. Acesso em: 06-06-2016.
- SÁ, G. D. C. Software lúdico de apoio ao ensino aprendizagem da matemática para crianças com discalculia. **REPOSITÓRIO DE RELATÓRIOS-Sistemas de Informação**, n. 1, 2015.
- SILVA, F. R. A. d. Ambiente computacional interativo para auxílio do processo de ensino aprendizagem de matemática básica. 2009.
- SOARES, M. dos S. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação ISSN 1677-3071 doi: 10.5329/RESI**, v. 3, n. 1, 2004.
- SOMMERVILLE, I. **Engenharia de software**. [S.l.]: São Paulo: Addison Wesley, 2003.
- TAJRA, S. F. **Informática na educação: novas ferramentas pedagógicas para o professor da atualidade**. [S.l.: s.n.], 2001.
- TALMA, T. M. **DESENVOLVIMENTO DE SOFTWARE DE AUXÍLIO AO FLUXO E AO COMPARTILHAMENTO DE INFORMAÇÕES ADMINISTRATIVAS EM AMBIENTES EMPRESARIAIS**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DE JUIZ DE FORA, 2006.
- VALENTINI, C. B.; SOARES, E. M. d. S. Aprendizagem em ambientes virtuais: compartilhando idéias e construindo cenários. **E-book-Aprendizagem em Ambientes Virtuais**, 2010.

VANROSSUM, G.; DRAKE, F. L. **The Python Language Reference**. [S.l.]: Python software foundation Amsterdam, Netherlands, 2010.

VESSELINOV, R.; GREGO, J. Duolingo effectiveness study. **City University of New York, USA**, 2012.

WOOLLEY, D. R. Plato: The emergence of online community. **Computer-Mediated Communication Magazine**, v. 1, n. 3, p. 5, 1994.

APÊNDICE A - DOCUMENTO DE REQUISITOS

1 Introdução

Este documento especifica os requisitos do sistema AskMath, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes.

1.1 Visão geral do documento

O documento apresenta os requisitos funcionais, onde para cada um destes define seu caso de uso e sua descrição detalhada. Além disso, define sua prioridade, explicado no tópico abaixo, e suas pré e pós condições, entradas e saídas caso existam. Logo em seguida, será apresentado os diagramas de caso de uso do sistema e alguns caso de usos críticos, e por fim o grafo de rastreabilidade entre os requisitos.

1.2 Prioridades dos requisitos

Cada requisito terá uma determinada prioridade, essa prioridade ira ajudar a equipe de desenvolvimento na escolha de quais requisitos mais se preocupar quando tiver desenvolvendo o sistema, criamos para isso, três níveis prioridades:

- Essencial: requisito é o requisito ao qual o sistema não entra em funcionamento sem ele. Esses requisitos tem que ser implementados o mais rápido possível.
- Importante: requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Esse requisitos deverão ser implementados, mas caso não sejam, o sistema ainda poderá funcionar consideravelmente.
- Desejável: requisito que não compromete o sistema, esse tipo de requisito é comumente deixado para versões posteriores do sistema.

2 Requisitos Funcionais

[RF001] Tipos de Usuário e Autenticação

O Sistema deverá possuir quatro tipos de Estudantes, Bolsistas, Professores e Administradores, e todos deverão ser autenticados; para isso, o mesmo deverá possuir um procedimento de autorização de utilizadores, onde cada utilizador se deve identificar através de um nome de usuário e uma senha. Apenas os utilizadores autorizados dessa forma podem acessar o sistema. Prioridade: Essencial

[RF002] Cadastro

Alunos podem se cadastrar no sistema. Quando o usuário não possuir nenhum código especial para se cadastrar, a única opção disponível para ele será de Aluno. Prioridade: Essencial

[RF003] Recuperar Senha

O Sistema deverá propiciar aos usuário uma opção para recuperar sua senha caso necessite. Prioridade: Essencial

[RF004] Hierarquia do Conteúdos

No sistema, existem disciplinas, lições para as disciplinas e problemas para as lições. Os professores poderão cadastrar várias disciplinas, e para cada disciplina, poderá ser cadastradas várias lições. Quando o Aluno entrar no sistema, ele poderá ver apenas as lições de uma determinada disciplina por vez, podendo alternar entre disciplinas. Os bolsistas irão adicionar problemas as lições, sendo assim, para poder um aluno responder uma lição, ele terá de escolher uma disciplina, depois uma lição e por fim a problema. Prioridade: Essencial

[RF007] Estrutura do Problema

Cada problema poderá possuir vários itens. Os problema serão somente de múltipla escolha, ele poderá possuir no mínimo dois itens e no máximo cinco itens, essa quantidade fica a critério do bolsista que a está cadastrado no sistema. Prioridade: Essencial

[RF008] Manter Disciplinas

Professores poderão adicionar, editar e excluir disciplinas e lições.
Prioridade: Essencial

[RF009] Manter Problemas

Assistentes poderão adicionar, editar e excluir problemas das lições.
Prioridade: Essencial

[RF0010] Alternar entre Disciplinas

Quando o aluno entrar no sistema pela primeira vez, ele verá uma lista com as disciplinas disponíveis, ele então deverá escolher uma opção, em seguida irão aparecer apenas lições referentes aquela disciplina, mas caso ele queira, ele poderá trocar facilmente de disciplina.
Prioridade: Essencial

[RF0011] Ver Detalhes da Lição

Antes de começar a resolver os problemas de determinada lição, o aluno deverá ver informações daquela lição.
Prioridade: Essencial

[RF0012] Sair da Lição

O Aluno poderá sair da lição antes mesmo de tê lá concluído, voltando depois da posição onde parrou. Ao sair, o aluno deverá ver estatísticas referentes a evolução dele na lição.
Prioridade: Essencial

[RF0012] Saltar Problemas

O sistema deverá permitir ao aluno, saltar problemas e rever os saltos realizados, isso com algumas restrições de quantidade de vezes.
Prioridade: Essencial

[RF0013] Pedir ajuda

Para todo problema, o sistema deverá apresentar um botão de ajuda. Ao clique do Aluno, o sistema deverá perguntar ao aluno se ele deseja obter ajuda do sistema ou de alguém que já concluiu aquela lição com bom aproveitamento. Caso o aluno opte pela primeira opção, o sistema exibirá um texto de ajuda cadastrado pelo assistente durante a criação do problema. Já no caso dele escolher a segunda opção, o sistema irá notificar aos alunos que estiverem online no momento que já concluirão aquela lição com certa pró-eficiência e caso algum deles aceite o pedido de ajuda, os dois encontram numa sala de bata-papo para conversar.

Prioridade: Essencial

[RF0014] Alterar Visibilidade do Problema

Um problema pode ser visível aos alunos ou não, quando o assistente estiver adicionando um problema, terá um campo marcado por padrão como verdadeiro que indicará se o problema estará visível ou não para os alunos, caso o mesmo não queira que o problema adicionado fique imediatamente visível aos alunos, ele desmarcará esse campo e futuramente ele poderá editar o problema e marca-lo quando quiser que os alunos possam acessá-la.

Prioridade: Essencial

[RF0015] Deficiências

Cada item incorreto de um problema, deverá linkar para uma deficiência, essa deficiência é outra lição. Pode ser possível, a partir de um item respondido incorretamente em um problema, identificar a deficiência do aluno, por isso, é fundamental que no momento que o assistente estiver adicionando os itens do problema, ele poderá escolher para os itens incorretos, uma ou várias outras lições, essas lições ficarão sendo as supostas deficiências do aluno caso ele erre aquele problema respondendo aquele item.

Prioridade: Essencial

[RF0016] Recompensas e Punições

A cada 3 problemas que o aluno responder corretamente, ele será parabenizado e ganhará um prêmio e a cada 3 errados ele será penalizado. Quando o aluno resolver seguidamente três problemas corretamente, os seus pontos acumulados irão dobrar e a cada vez que ele errar três questões seguidamente, seus pontos acumulados serão subtraídos em 25

Prioridade: Essencial

[RF0017] Fórum

Caso o aluno possua alguma dúvida, o sistema deverá possuir um fórum onde esse aluno poderá postar suas dúvidas, para que professores, bolsistas ou outros alunos possam lhes ajudar com o problema, nesse fórum, deve possuir tópicos, comentários para os tópicos e nos comentários devem oferecer uma opção para ele comentar com imagem. Prioridade: Essencial

[RF0018] Ordenar Problemas

O Sistema deve apresentar ao bolsista uma forma simples para ele ordenar a sequência dos problemas de cada lição. Prioridade: Essencial

[RF0018] Suporte a Latex

O Sistema deve permitir ao bolsista adicionar código Latex(funções) na criação dos problemas e lições. Quando os bolsista adicionarem os problemas, o sistema devera reconhecer código Latex referente a fórmulas matemáticas. Toda vez que o bolsista colocar um código Latex entre as tgs '\$', o sistema deverá reconhecer isso, e mostrar para o usuário a imagem da fórmula referente aquele código. Prioridade: Essencial

3 Requisitos Não Funcionais

[RN001] Tecnologias

O Sistema deve ser desenvolvido apenas com tecnologias open source. Prioridade: Essencial

[RN002] Persistência dos Dados

O sistema deverá utilizar como sistema de gerenciamento de banco de dados o PostgreSQL. Prioridade: Importante

[RN003] Segurança

O sistema não apresentará aos usuários quaisquer dados de cunho privativo. Prioridade: Essencial

[RN004] Estrutura

O sistema deverá ser desenvolvido de forma modularizada, para permitir a reusabilidade de tais módulos por outras aplicações no futuro.
Prioridade: Essencial

[RN005] Padrões

O Sistema deverá ser desenvolvido utilizando os princípios de Orientação a Objetos.
Prioridade: Essencial

[RN006] Ambiente de Execução

O sistema deverá ser acessado completamente via browser HTTP/HTML.
Prioridade: Essencial

[RN007] Acessibilidade

O Sistema devera possuir Responsive Web Design, assim como as várias diretivas de acessibilidade para web.
Prioridade: Importante

[RN008] Internacionalização

O Sistema será disponibilizado em inglês, mas de forma a permitir que versões em línguas latinas possam ser produzidas sem necessidade de ter acesso ao código fonte.
Prioridade: Importante

[RN009] Desempenho

Quando um aluno responder um problema, a correção deverá ser apresentada ao aluno, no máximo, em 2 segundos.
Prioridade: Essencial

[RN010] Log do Sistema

O Sistema devera salvar um histórico de todas as ações que os usuários realizarem.
Prioridade: Essencial

[RN010] Log do Sistema

O Sistema devera salvar um histórico de todas as ações que os usuários realizarem.
Prioridade: Essencial

4 Requisitos de Domínio

[RN001] Log do Sistema

O Sistema devera salvar um histórico de todas as ações que os usuários realizarem.
Prioridade: Essencial

5 Casos de uso

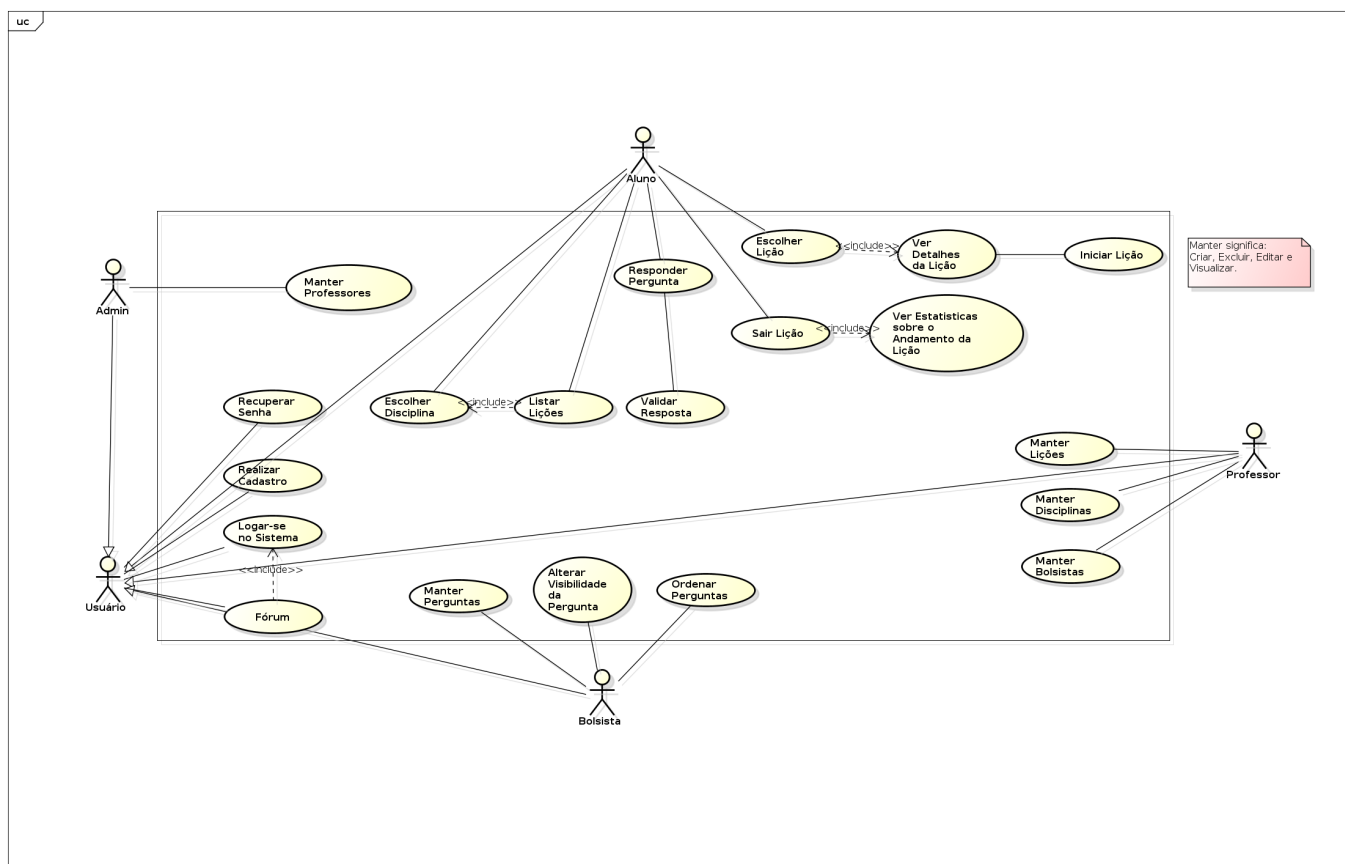


Figura 1 – Diagrama de Casos de Uso

5.1 Descrição dos Casos de Uso

- **Manter Professor:** O Administrador terá uma opção onde poderá manter professores: isso funcionará da seguinte forma: Para cadastrar, o Administrador logado, gera um código de alguns caracteres e dá ao professor, o professor de posse desse código, entra na tela de cadastro do usuário e escolhe o tipo de usuário que ele quer se cadastrar e informa o código de acesso que ele possui, assim ele terá permissão para se cadastrar como Professor.
- **Manter Assistente:** De forma análoga ao Administrador, o professor também irá gerar um código e entregar ao assistente, o assistente de posse desse código irá entrar na tela de cadastro do sistema, informar o tipo de usuário e o código de acesso.
- **Recuperação de Senha:** No momento do cadastro, cada usuário deverá indicar um e-mail válido, caso futuramente o usuário necessite alterar a senha, um e-mail será enviado à esse e-mail cadastrado com um link onde o mesmo poderá acessar para recuperar a senha.
- **Manter Conteúdos:** Apenas os professores terão permissão para manter conteúdos e lições, ao adicionar uma lição, ele terá que informar apenas o nome da lição e ao adicionar uma lição, ele terá que informar os Pré-Requisitos (Outras lições que eles recomendam ter sido concluídas para se prosseguir na atual) e Sugestões de Estudo (outras lições que eles recomendam seguir após concluir essa lição) para aquela lição assim como a quantidade máxima de pulos que o aluno poderá realizar naquela lição.
- **Manter Problemas:** Cada lição possuirá uma lista de problemas e os mesmos serão adicionadas pelos assistentes. Quando um assistente adicionar um problema, ele terá de informar a qual lição que ele pertence, os itens que ela terá, assim como o item correto, a ajuda caso esse problema necessite ter, a quantidade de pontos que ele terá e se ele irá ficar visível imediatamente para os alunos ou não.
- **Ver Detalhes da Lição:** Quando o aluno optar por responder as perguntas de uma determinada lição, antes de mais nada, ele precisa saber os detalhes daquela lição, detalhes do tipo: Quantidade de questões, Máximo de Saltos Permitidos, Pré-Requisitos, Curiosidades sobre o conteúdo daquela lição.
- **Sair da Lição:** Enquanto o aluno estiver resolvendo os problemas, o sistema deverá lhe oferecer a opção dele sair da lição, quando ele sair, ele terá de ver as estatísticas referentes ao andamento dele naquela lição como: Acertos, Erros, Saltos, Quantidade de Pontos Acumulados e também uma lista com as lições sugeridas.

- Ordenar Problemas: Quando o bolsista adicionar um problema, ela será adicionada automaticamente logo depois dos outros problemas daquela lição (se pensarmos numa lista), mas ele deverá ter uma opção onde apenas arrastando os problema de posição ele possa reordena-los, utilizando apenas o mouse.

APÊNDICE B - DOCUMENTO DE PERSONAS

1 Charlie - Técnico em Informática



Figura 1 – Imagem de Charlie

Empresa: Ele trabalha na TechSolutions, uma empresa pequena, mas está ganhando clientes e ampliando seus negócios.

Idade: 24 anos.

Genêro: Masculino.

Educação: Ensino técnico.

Mídias: Lê a revista Exame e usa ativamente o email para trocar informações com outros funcionários da empresa.

Objetivos: Como a empresa que Charlie trabalha está ampliando os negócios, ela necessita que alguns de seus funcionários possuam um melhor conhecimento na área, por isso, Charlie decidiu investir num ensino superior, seu objetivo é concluir seu curso de Engenharia de Software e voltar a trabalhar normalmente para sua empresa.

Desafios: Charlie está com muita dificuldade na primeira disciplina de Matemática de seu curso, como ele já concluiu o ensino médio há bastante tempo, o mesmo não se lembra mais de como resolver questões simples de matemática como funções polinomiais, racionais e trigonométricas. Ele começou a estudar, mas não consegue encontrar uma forma intuitiva para verificar se está indo bem nos seus estudos, ficando preso aos exercícios que ele encontra nos livros.

Como minha empresa pode ajudá-la: O AskMath possibilitará que Charlie estude todos os conteúdos que ele tinha visto no ensino médio e já os tinha esquecido, são lições agrupadas por disciplina, cada lição possui um conjunto de questões e vídeo aulas criadas

especialmente para pessoas com dificuldade nessas disciplinas. Para as questões, nosso sistema possibilitará que Charlie veja instantaneamente se sua resposta é correta ou não, ele também poderá pedir ajuda e saltar uma questão o mesmo não se sinta a vontade para respondê-la naquele momento. Com isso é possível que ele acompanhe através de estatísticas como está seu desempenho no sistema.

2 Ruby - Bolsista de Graduação

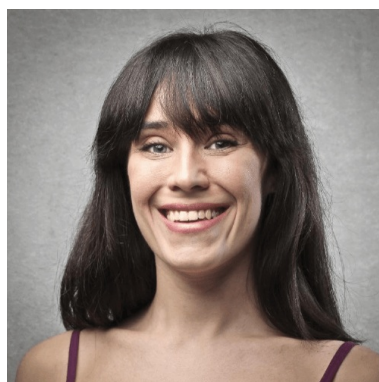


Figura 2 – Imagem de Ruby

Empresa: Ela trabalha como bolsista de monitoria das disciplinas envolvendo Matemática na Universidade Federal do Ceará.

Idade: 21 anos.

Genêro: Feminino.

Educação: Ensino superior.

Mídias: Usa ativamente o facebook, twitter e wattsapp.

Objetivos: O principal objetivo de Ruby é terminar sua graduação e tentar uma bolsa de mestrado numa universidade renomada.

Desafios: Ruby como monitora das disciplinas de Matemática, não consegue se aproximar dos alunos para tirar suas dúvidas, segundo ela, "eles tem medo de dizer para nós monitores que está com dificuldade", então Ruby está em busca de novas formas de ajudar os estudantes que estão com dificuldades.

Como minha empresa pode ajudá-la: Com o AskMath, Ruby poderá ajudar esses alunos, ela poderá adicionar questões para os mesmos exercitarem seus conhecimentos.

3 Samuel - Professor



Figura 3 – Imagem de Samuel

Empresa: Ele trabalha como professor da disciplina de Matemática Básica na Universidade Federal do Ceará.

Idade: 59 anos.

Gênero: Masculino.

Educação: Doutorado.

Mídias: Lê o jornal The New York Times e usa email para tirar dúvidas dos alunos
Objetivos: Samuel é um professor exemplar, se preocupa muito em ensinar seus alunos, seu principal objetivo é ensinar da melhor forma possível seus alunos, de forma que todos possam seguir excelentes carreiras quando se formar e se tornam ótimos profissionais.

Desafios: Samuel gosta de acompanhar o andamento dos estudos de seus alunos, entretanto, quando ele os questiona sobre alguma dificuldade que eles estejam enfrentando, os mesmos dizem que está tudo bem, o que não reflete em suas notas. Sabendo disso, Samuel fica triste, já que não consegue saber como anda o nível de aprendizado de sua turma e gostaria de ajudá-los mais.

Como minha empresa pode ajudá-la: Com o AskMath, Samuel pode acompanhar o andamento de sua turma, e saber em quais conteúdos eles estão com mais dificuldade, para ele poder focar mais neles. Samuel também poderá adicionar ele mesmo os conteúdos que ele achar mais interessantes para seus alunos, além disso, ele verá as dúvidas que eles postam no fórum e poderá ele mesmo responder a essas dúvidas.

4 Stefane - Estudante do Ensino Médio



Figura 4 – Imagem de Stefane

Empresa: Ela trabalha no salão de beleza de sua mãe.

Idade: 18 anos.

Genêro: Feminino.

Educação: Ensino médio.

Mídias: Usa ativamente o facebook e WatssApp.

Objetivos: Stefane busca concluir o ensino médio e tirar uma boa nota no ENEM para tentar conseguir uma vaga para cursar Sistemas de Informação na UFC, que é o curso de seus sonhos.

Desafios: Stefane, como estudante, não gosta de estudar matemática por livros, ela acha incomodo ter que levar seus livros enormes de matemática em sua bolsa para o salão de beleza de sua mãe, mas ela quer ficar estudando enquanto não aparece clientes no salão. Um outro problema de Stefane, é que às vezes, ela fica com dúvida na resolução de alguma questão e precisa ligar para seus amigos em busca de explicações que nem sempre encontra. Stefane também sente uma certa dificuldade em aprender através de papéis, ela acha que nada é melhor do que a explicação visual de uma pessoal.

Como minha empresa pode ajudá-la: Com o AskMath, Stefane não precisará mais levar livros de Matemática em sua bolsa, para aprender matemática ela precisa apenas de uma celular, isso possibilita que ela possa continuar estudando no salão quando não tiver muitos clientes. Para solucionar as dúvidas de Stefane, temos pessoas qualificadas esperando para ajudá-la, ela poderá publicar suas dúvidas num fórum onde essas pessoas podem responder, assim como outros pessoas, assim como Stefane, mas que já tenha tido a mesma dúvida. Como Stefane não gosta de papéis, o AskMath possui um conjunto de vídeo aulas criadas por pessoas capacitadas, que sabem a melhor forma de ensinar através de recursos visuais.

APÊNDICE C - DOCUMENTO DE ARQUITETURA

1 Introdução

Neste documento será detalhado a arquitetura do sistema proposto para o projeto AskMath. Como o projeto é voltado para a web, o sistema é formado por diversos padrões de projetos de mercado, principalmente, padrões orientados a objetos. Vamos destacar cada parte da arquitetura escolhida para a fácil compreensão de outras equipes que venham a trabalhar no projeto, seguindo a risca seu modelo para permitir que o sistema seja facilmente modificado ou incrementado sem que a estrutura do software seja perdida.

2 Objetivos

- Prover uma visão geral da arquitetura do projeto, detalhando cada parte da estrutura do sistema para permitir a compreensão do mesmo.
- Permitir que este documento seja utilizado por outras equipes de desenvolvimento que estão dando continuidade ao projeto, quanto a novos integrantes inseridos na equipe.
- Permitir que este documento sirva como meio de comunicação entre o Arquiteto de Sistema e a equipe desenvolvedores.
- Apresentar aos *stakeholders* uma visão de alto nível de como o sistema é implementado e como esta estruturado.
- Aumentar o desempenho e a robustez, bem como a capacidade de distribuição e manutenibilidade do sistema.

3 Considerações Gerais

As definições de arquitetura do processo de desenvolvimento do software esta preocupado como o sistema deve ser organizado e com a estrutura geral do sistema, estas definições tem que atender as especificações do projeto, desde as especificações de segurança, regras de negócio, até a parte de persistência de banco de dados. As decisões de projeto de arquitetura têm efeitos profundos sobre a possibilidade de o sistema atender ou não aos requisitos críticos, como desempenho, confiabilidade e manutenibilidade.

As definições da arquitetura atendem as especificações de projeto documentadas até o presente momento, apresentando um modelo completo do sistema que mostre seus diferentes componentes, suas interfaces e conexões.

4 Responsabilidades

Toda a equipe de desenvolvimento é responsável por elaborar este documento e por manter a integridade do mesmo durante o processo de desenvolvimento. Cada membro deve:

- Analisar todas as mudanças arquiteturais significativas e documentá-las;
- Ao verificar uma possível alteração na arquitetura, convocar uma reunião com toda a equipe para discutir a possível alteração.

5 Arquitetura

A arquitetura foi desenvolvida para ser de baixo acoplamento e que ao mesmo tempo fosse independente de tecnologias existentes do mercado, sendo assim, poderíamos por exemplo futuramente trocar o framework Django pelo o Spring, e o mesmo seria transparente para os programadores e para o sistema .

5.1 Elementos que compõe a Arquitetura

A arquitetura é composta por elementos, em que em conjunto produzem o produto final. Esses elementos são: Database, Models, Views e Templates.

Nos tópicos a seguir descrevemos cada um dos componentes e o seu papel dentro da arquitetura como um todo, além de discutirmos a tecnologia e os padrões adotados para a implementação dos mesmos.

5.1.1 Database

No desenvolvimento de sistemas, necessitamos muitas vezes salvar as informações geradas para eventuais utilizações das mesmas, para isso, temos disponíveis os banco de dados. Em geral, as linguagens de programação nos proporcionam formas de acessar esses dados, porém de forma muito complexa que acaba ocasionando um alto acoplamento para termos de alta coesão. O framework que utilizaremos, nós proporciona uma camada apenas de banco de dados ou camada de persistência.

Nessa camada estão as entidades do sistema, e as mesmas implementam funcionalidades de conexão e outros controles que são iguais a todos os tipos de acessos a bancos de dados.

Com esse tipo de acesso aos dados, conseguimos trocar facilmente de banco de dados sem afetar a implementação do sistema. Por exemplo, caso estejamos utilizando um sgbd relacional como o postgresql para armazenar os dados e necessitarmos migrar para

um sgbd não “relacional”, os famosos banco de dados NoSQL, como o MongoDB, não será necessário alterar nenhum aspecto da implementação, pois as regras de negócio vão está totalmente separadas das funcionalidades de acesso ao banco de dados.

5.1.2 Models

São as entidades do sistema. Entidades são abstrações do mundo real modelados em forma de tabelas que guardarão informações no banco de dados. Nessa arquitetura, os modelos contemplarão também as associações entre as entidades, sendo que cada entidade também terá além de seus atributos, um conjunto de métodos para se relacionar com o sistema.

Os modelos implementarão um ORM(Object-Relational Mapping) para as views acessarem o banco de dados, tornando o sistema independente do acesso ao banco.

5.1.3 Views

Todo sistema é formado por um conjunto de regras de negócio. Uma regra de negocio é fluxo lógico que deve ser processado para que seja gerado um resultado válido.

Uma View é responsável pela execução de um ou mais fluxos de execução que são modelados em um caso de uso, ou seja, uma visão é uma implementação da regra de negocio, elas são basicamente funções que aceitam como primeiro parâmetro uma *request* que representa uma requisição Web (além de outros parâmetros) vinda do usuário pelo *browser*, ela irá tratar essa requisição e retornar uma resposta ao usuário.

Dessa forma, cada caso de uso acaba tornando se uma view, sendo que views podem dispor de outras views para realizarem suas funções.

5.2 Desenho geral da arquitetura

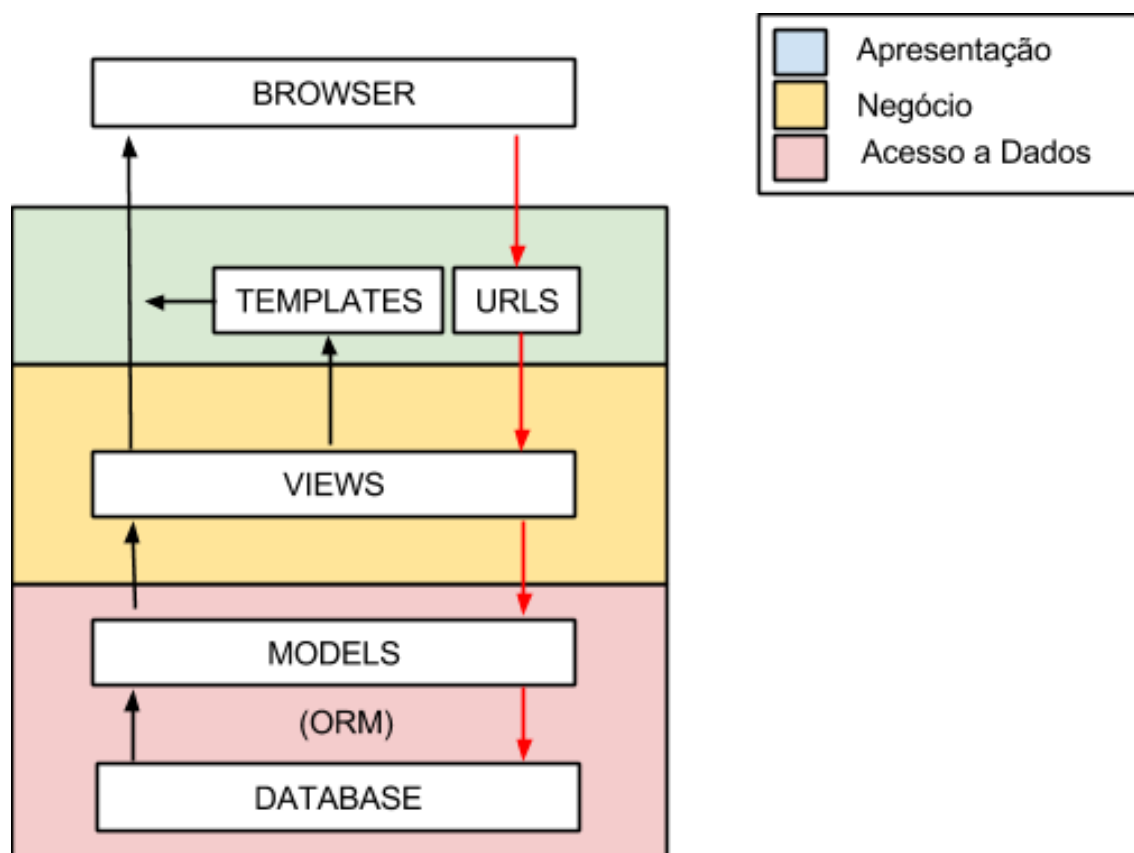


Figura 1 – Desenho Geral da Arquitetura

Nessa arquitetura, o browser do cliente utiliza de uma URL para realizar uma requisição numa view, essa view pode utilizar os models para extrair dados do banco de dados e ela própria criar um template ela mesma retornar como uma forma de resposta ao usuário, ou então ela pode utilizar um template pronto apenas moldar esses dados dentro de um template pronto e retornar isso ao browser do cliente.

6 Padrões de projeto

Um padrão de projeto representa o trabalho de uma pessoa que encontrou o mesmo problema, tentou muitas soluções possíveis, selecionou e descreveu uma das melhores e você deve se aproveitar disso.

O conhecimento de padrões permite decidir o que deve ser feito e o que deve ser evitado, sistemas baseados em padrões têm mais qualidade. Para o AskMath, foram analisados alguns padrões de projeto e selecionados aqueles que poderiam ser satisfatoriamente aplicados.

6.1 Proxy

Fornecer um substituto ou marcador da localização de outro objeto para controlar o acesso a esse objeto, podemos controlar o acesso aos métodos de uma entidade da seguinte forma: cria-se uma classe EntidadeProxy que estende de Entidade, e reimplementa cada um dos métodos, nessa nova implementação, você solicita que o usuário informe uma autenticação que possua acesso, ao informar, esse método irá apenas chamar o método da Entidade.

6.2 Chain of Responsibility

Evitar o acoplamento do remetente de uma solicitação ao seu receptor, ao dar a mais de um objeto a oportunidade de tratar a solicitação. Encadear os objetos receptores, passando a solicitação ao longo da cadeia até que um objeto a trate.

7 Objetivos e Restrições Arquiteturas

Apresentaremos aqui os requisitos e objetivos do software que têm algum impacto na arquitetura, tais como: segurança, proteção de dados, privacidade, portabilidade, distribuição, reuso. Também são descritos nesta seção restrições arquiteturais que se aplicam ao projeto, tais como: estratégias de modelagem e implementação, ferramentas de desenvolvimento, sistemas legados.

7.1 Requisitos básicos

- Ubuntu Server como sistema operacional de produção.
- Utilização apenas de componentes open-source.
- PostgreSQL como sistema de gerenciamento de banco de dados.
- Django como framework de desenvolvimento.
- O sistema deverá ser Web.

7.2 Estratégias de implementação

- Persistência de tipagem (ex: formato de CPF, Data) devem ser feitos tanto no cliente através de Javascript, como no servidor através de regex(Regular Expression).
- Persistência de obrigatoriedade, verificar se todos os campos obrigatórios foram preenchidos deve ser feito tanto no cliente via javascript como no servidor e nas

duas opções deve-se apresentar alertas caso algum campo seja enviado vazio para a requisição.