DESENVOLVIMENTO DO SISTEMA DE GERENCIAMENTO DE UNIDADE ESCOLAR - SIGUE

Marlos Cézar Da Silva Nascimento[1]
Romério Ribeiro Da Silva[2]
Sérgio Augusto De Souza Moraes[2]

RESUMO

No cenário global atual, que está envolto por uma globalização irrenunciável e que trouxe consigo um reflexo muito forte sobre as organizações, principalmente quanto às incertezas que rondam as organizações acerca do mercado, obriga-se que estas estejam cada vez mais flexíveis e inovadoras, o que requer que elas estejam, de fato, abertas à mudança, uma vez que o ecossistema dos potenciais concorrentes se elevou à dimensão global. Diante do exposto, à medida que cada organização busca alçar voos maiores e conseguir fatias maiores do mercado, é necessário que elas adotem estratégias eficazes que resultem em atividades que estejam voltadas para o crescimento da empresa como um todo. Assim, a organização deve possuir um conjunto de informações úteis e necessárias que possibilite uma tomada de decisão mais acertada possível e para isso contam com os recursos de tecnologia da informação e comunicação. A partir dessas informações, busca-se desenvolver um trabalho que parte de uma necessidade comum entre as escolas de ensino básico, principalmente as do setor público, que necessitam de uma agilidade maior na realização das tarefas administrativas e uma melhor forma de guardar e buscar informações dos alunos, reduzindo a grande quantidade de papéis que são manuseados e produzidos durante as atividades do dia a dia, e assim exibe-se o desenvolvimento do sistema de gerenciamento de unidade escolar (SIGUE). O trabalho tem como objetivo contribuir para a melhoria das atividades administrativas da instituição de ensino básico por meio do desenvolvimento de um *software*. Dessa maneira apresenta-se que a utilização do sistema desenvolvido, a partir deste trabalho, pelas instituições de ensino poderá contribuir para que elas possam gerir informações de maneira organizada, melhorando os processos organizacionais, haja vista que a informatização hoje é

[1] Acadêmico do curso de Sistemas de Informação – Faculdade Atenas
[2] Docente do curso de Sistemas de Informação – Faculdade Atenas

um fator crítico de sucesso para as mesmas.

Palavras-chave: Tecnologia da informação e comunicação. Tomada de decisão. Gerenciamento de unidade escolar.

ABSTRACT

In today's global scenario, which is surrounded by a globalization that cannot be renounced and which has brought with it a very strong reflection on the organizations, mainly regarding the uncertainties that surround the organizations about the market, forcing them to be more flexible and innovative, which requires that they are, in fact, open to change, as the ecosystem of potential competitors has risen to the global dimension. In view of the above, as each organization seeks to raise larger flights and achieve larger market shares, it is necessary for them to adopt effective strategies that result in activities that are geared towards the growth of the company as a whole. Thus, the organization must have a set of useful and necessary information that allow the best decision-making possible and, for this, they rely on the resources of information technology and communication. From this information, it is sought to develop a work that starts from a common need among the elementary schools, especially those of the public sector, that needs a greater agility in the accomplishment of the administrative tasks and a better form to store and to look for information of students, reducing the large amount of paperwork that is handled and produced during day-to-day activities, and thus demonstrates the development of the school unit management system (SIGUE). The work aims to contribute to the improvement of the administrative activities of the basic education institution through the development of software. In this way, it is presented that the use of the software developed from this work, by the educational institutions, can contribute so that they can manage information in an organized way, improving the organizational processes, since computerization today is a critical success factor for them.

***Keywords**: Technology of Information and communication. Decision making. School unit management.*

INTRODUÇÃO

As organizações da atualidade estão em uma constante transformação de seus processos e formas de trabalho, permitindo a percepção da crescente utilização de recursos por meio da *web*. Neste contexto, é possível observar que as operações empresariais estão tendenciadas a migrar suas atividades e serviços locais para atividades cada vez mais incorporadas ao mundo virtual, pois, segundo Abreu e Machado (2012), no século atual, possuir apenas dados e informação já não é o bastante, pois as demandas são altas e requer velocidade de atendimento ainda mais rápidas. Esta visão ficou conhecida por *on demand*, em que a conectividade, ou comunicação, se tornou o ponto principal.

De acordo com a visão de Escrivão Filho, Moraes e Terence (2004), a tecnologia da informação nos dias atuais passou a ser um elemento imprescindível para a competitividade das organizações, sendo um fator capaz de gerar diferenciação entre as empresas, uma vez que as mesmas, para se manter competitiva, devem acompanhar as principais tendências do seu ambiente externo.

Segundo Albertin e Albertin (2008), o emprego da tecnologia da informação pelas organizações abre um leque de oportunidades substanciais para as mesmas, à medida que seus benefícios são aproveitados de maneira correta; entretanto, a dinâmica do ambiente em que a empresa está inserida deve ser compreendida detalhadamente, superando seus desafios, para que se possa alcançar um bom desempenho empresarial.

Potter, Rainer Jr e Turban (2005) expõem que independente da organização, com ou sem fins lucrativos, pública ou privada, estão imersas na economia digital no século XXI, em que se cria uma plataforma global que possibilita a interação entre pessoas e organizações, em um rol que abrange a comunicação, colaboração e busca por informações; grandes fatores geradores de competitividade e vantagens estratégicas.

Nesta ótica, a tecnologia da informação e comunicação é um fator primordial para as organizações inteligentes, por fornecer suprimentos indispensáveis para a gestão dos negócios, como os sistemas de informação, que geram e gerenciam conhecimento de forma organizada. Abreu e Machado (2012, p. 43) apresentam que “o que realmente importa é que alguém vai assumir a responsabilidade de entregar algumas funções de TI, como serviços, para

alguns clientes e eles não precisam saber como funciona; simplesmente vão usar". A partir dessa realidade, observa-se também que as instituições de ensino estão em busca de atividades mais integradas para atender às suas necessidades e para que possam garantir uma maior agilidade e controle sobre o gerenciamento administrativo escolar, bem como a informatização dos processos de cadastramento, manipulação e consulta por informações.

Assim, busca-se no desenvolvimento do presente trabalho apresentar o desenvolvimento do sistema de gerenciamento de unidade escolar, denominado SIGUE, o qual será desenvolvido na plataforma *Web*, com a linguagem Java, utilizando a tecnologia JSF e o *framework* Bootsfaces.

TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NAS ORGANIZAÇÕES

A capacidade competitiva das organizações atualmente é fator de grande importância e requer bastante esforço, por isso, Sacilotti (2011, p. 12), apresenta que "o grande desafio dos administradores tem sido manter a capacidade competitiva de suas empresas no Mercado". Assim, é requerido que sejam criadas estratégias capazes de impulsionar o poder das empresas frente aos desafios encontrados cotidianamente. Para enfrentar os obstáculos presentes no caminho, ainda de acordo com o auto, é necessário que as empresas se organizem de modo a dar o devido tratamento às informações importantes para o negócio, sendo este o principal ponto a se levantar em se tratando de capacidade e vantagem competitiva. Sacilotti (2011, p. 12) preconiza que "a tecnologia, então, tornou-se um elemento- chave, que tem auxiliado no processo de diferenciação de mercado e destacado favoravelmente as empresas frente à concorrência". Carvalho *et al.* (2001) cita que o emprego da tecnologia da informação contribui com a viabilização de novas estratégias organizacionais.

Notando-se assim, a partir dessas informações, um cenário econômico mundial atual que está em constante mudança, fruto de uma intensa globalização. Os administradores devem estar atentos e abertos a mudanças para que possam adequar as organizações às demandas do mercado e mantê-la competitiva. Carvalho *et al.* (2001) observa que "o uso eficaz da TI e a integração entre sua estratégia e a estratégia do negócio vão além da ideia de ferramenta de produtividade, sendo muitas vezes fator crítico de sucesso". Sacilotti (2011) aponta

que neste momento a tecnologia atuará de forma a disponibilizar contribuições importantes para a tomada de decisão.

De acordo com Morales e Rossetti (2007), a tecnologia da informação e comunicação tem sido uma ferramenta de comunicação e gestão, refletindo na capacidade das pessoas e organizações se manterem competitivas em seu mercado de atuação. O autor analisa ainda o fato de que a tecnologia da informação e comunicação deve estar ligada a todas as atividades da empresa, atuando como uma ferramenta de apoio à incorporação do conhecimento e servindo como o principal fator agregador de valor aos produtos, processos e serviços.

SISTEMA DE INFORMAÇÃO

Atualmente as organizações estão em uma disputa muito acirrada para se manter no mercado devido a demasiada aceleração do ambiente econômico, nesse sentido, de acordo com Abreu e Machado (2012), um dos fatores que motiva esta intensa mutação deste ambiente é, sem dúvida, a globalização. Desse modo muitos detalhes devem ser considerados na gestão organizacional, como, por exemplo, a valorização e gerenciamento da informação advinda das diversas origens, sendo elas internas ou relativas ao ambiente em que a empresa está inserida.

Seguindo o raciocínio de Andrade, Audy e Cidral (2005), a tecnologia da informação é caracterizada como um grande diferencial competitivos da atualidade, facilitando e auxiliando a tomada de decisão. Um sistema de informação, ainda de acordo com o mesmo autor, é um mecanismo que visa proporcionar à organização informações úteis, necessárias e confiáveis, por meio de coleta, processamento, armazenamento, distribuição dos dados e posterior relacionamento e contextualização, para que ela possa atuar em um determinado ambiente.

Para o eficaz tratamento das informações necessárias à vida da organização, segundo Abreu e Machado (2012), hoje utiliza-se mecanismos tecnológicos proporcionado pela informática, como os sistemas de informação, cada um para uma solução específica, como o Sistema de informação executiva – SIE, Sistemas de apoio às decisões – SAD, Recursos de Inteligência Artificial – IA entre muitos outros.

Na visão de Bio (2008, p 22), “um sistema de informação é uma rede de subsistemas em que cada qual se decompõe em procedimentos que coletam dados, os processam e produzem e distribuem as informações resultantes”. Analisando esta abordagem, pode-se perceber que um sistema de informação é responsável por acionar e controlar os processos em seu ciclo de vida, o que incidirá sobre a eficiência operacional da organização. Segundo Potter, Rainer Jr e Turban (2005) os sistemas de informação, ou sistemas de informação baseado em computador, (SIBC) são direcionados de acordo com a amplitude de suporte: funcionais ou departamentais, em que giram em torno de departamentos ou funções tradicionais da empresa, como contabilidade ou manufatura, por exemplo; Corporativos, que compreende os interdepartamentais e os organizacionais, possibilita a comunicação entre as pessoas e o acesso à informação por toda a empresa, por exemplo os ERPs; Por fim, os Inter organizacionais que permitem a conexão entre organizações. Nesta perspectiva, Andrade, Audy e Cidral (2005, p. 110) visualizam um detalhamento dos sistemas de informação apresentando suas metas fundamentais: “Suporte ao controle e à integração dos processos de negócio e funções organizacionais, suporte ao processo decisório nos diversos níveis organizacionais e suporte a estratégias competitivas”. Outra forma de análise dos sistemas de informação, ainda de acordo com Potter, Rainer Jr e Turban (2005) é quanto ao suporte oferecido aos colaboradores de cada nível organizacional, os quais permeiam tanto na esfera operacional, auxiliando na tomada de decisão de curto prazo; tanto no nível tático, interligando a órbita operacional e a estratégica, auxiliando na tomada de decisões de médio prazo; e no nível estratégico, auxiliando nas decisões de longo prazo e permitindo que as decisões sejam tomadas com base em informações úteis e concretas acerca do contexto interno e externo da organização.

Sob este ponto de vista, percebe-se que tais artifícios tecnológicos cumprem uma função estratégica no negócio ao participar de todos os níveis organizacionais fornecendo informações do contexto interno e externo almejando o planejamento de longo prazo. “A evolução integrada de tecnologia da informação e comunicação, pessoas e gestão contribuiu para o desenvolvimento de organizações inteligentes”. (ABREU e MACHADO, 2012).

PROCESSOS DE DESENVOLVIMENTO DE *SOFTWARE*

Entende-se que o desenvolvimento de um sistema corresponde a um conjunto de atividades que resultam na produção de um produto de *software*, às quais dá-se se o nome de processos de *software*, que, de acordo com Ferreira *et al*. (2015), é uma das estratégias para se conseguir agregar qualidade ao software.

O desenvolvimento de software, de acordo com Cukierman, Prikladnicki e Teixeira (2007), envolve uma série de conhecimento das mais diversas áreas, sendo necessária então uma envolvente aproximação entre as ciências humanas e sociais, pois as atividades realizadas na criação de produtos de software vão além de processos puramente técnicos.

Segundo Sommerville (2011), estes processos podem acontecer de diferentes maneiras; entretanto, existem quatro atividades, ou processos, primordiais que são imprescindíveis para a engenharia de *software*, que são a especificação, projeto e implementação, validação e evolução. Já, para Azevedo Junior e Campos (2008), existem três fases genéricas: definição, desenvolvimento e manutenção. Comparando as ideias dos autores, percebe-se que a fase desenvolvimento de um dos autores corresponde à fase projeto e implementação e validação do outro autor, mas que em sua maioria os processos se equivalem, apresentando apenas definições diferentes, destacando-se assim a importância da compreensão das fases na execução de rotinas de desenvolvimento de sistemas de informação computadorizados.

A primeira atividade é a especificação de *software* ou engenharia de requisitos, segundo Sommerville (2011), ou definição, de acordo com Pressman (1995) e Azevedo Junior e Campos (2008). Nesta atividade são determinadas quais serão as funcionalidades que deverão ser implementadas, bem como suas respectivas restrições, de modo que satisfaça as necessidades dos *stakeholders.* Neste processo, encontram-se as atividades de estudo de viabilidade, análise de requisitos, especificação de requisitos e validação de requisitos.

A Segunda, consiste no projeto e implementação de *software,* de acordo com Sommerville (2011), ou desenvolvimento, para Pressman (1995) e Azevedo Junior e Campos (2008). É onde o *software* será projetado e construído com base nas especificações levantadas anteriormente; ou seja, transformam as especificações em um *software* executável. Abrange as atividades de projeto de

arquitetura, projeto de interface, projeto de componente e projeto de banco de dados.

A terceira atividade, como abordado por Sommerville (2011), é a validação de *software*. Nesta fase, é verificado se o que foi construído está de acordo com o que foi projetado para atender às necessidades do cliente. O principal método de validação é por meio de testes de programa. Possui os seguintes processos: testes de desenvolvimento, teste de sistema e teste de aceitação. Para Pressman (1995) esta atividade ainda se encontra na fase de desenvolvimento.

A quarta atividade fundamental consiste na evolução de *software*, para Sommerville (2011), ou manutenção, segundo Pressman (1995) e Azevedo Junior e Campos (2008), em que se altera o *software* produzido inicialmente, incluindo novas funcionalidades para satisfazes as novas necessidades do cliente. Envolve também a manutenção do sistema ao se realizar correções, adaptação e melhoramento funcional.

Sommerville (2011) esclarece que, embora os processos sejam complexos, não se pode dizer que existe um processo ideal. Muitas empresas criam seus próprios processos para desenvolverem *softwares*, o que permite que sejam melhorados; entretanto, existem organizações que não empregam e aproveitam os métodos da engenharia de *software*.

SOFTWARE

Para que os sistemas de computação possam ser executados de modo a solucionar problemas é necessário que as partes físicas sejam controlados por um sistema lógico. Estes sistemas lógicos são os sistemas ou *software*; entretanto, para Pressman (1995), ao contrário do hardware, o *software* não se desgasta com o tempo e o uso. Segundo Andrade, Audy e Cidral (2005, p. 171), *software* corresponde aos programas que que um sistema de computação consegue executar, e ainda considera que são desenvolvidos com vistas a solucionar um problema específico.

Esta solução pode englobar inúmeros sistemas, surgindo, então, um sistema de *software*. Dito isto, pode-se considerar que um *software* tem por fim auxiliar os usuários nas suas atividades diárias ou profissionais que podem ser

solucionadas ou facilitadas por um computador, uma vez que o mesmo autor ainda expõe que sistemas de *software* operam conjuntamente com vistas a obter resultados voltados para resolução de problemas de uma área específica. Pressman (1995) complementa explanando que dependendo do tipo de aplicação cada sistema se encaixará em uma categoria específica, como, por exemplo, os *softwares* básicos, que executam funções básicas dando apoiando outras aplicações; *software* comercial, que é responsável por facilitar as operações comerciais e a tomada de decisão; e várias outras categorias.

Sommerville (2003), em uma visão mais ampla, argumenta que sistema de *software* compreende um conjunto de programas e seus arquivos de configuração, bem como a documentação do sistema e a do usuário. Para Pressman (1995, p. 3) “o *software* é um fator que diferencia”, assim, sua utilização nos dias de hoje tornou-se algo essencial na vida das organizações, fornecendo uma identidade única a cada uma delas. Pressman (1995) cita que um sistema é criado levando-se em conta as exigências dos clientes. Tais exigências, ou necessidades, são escritas, por meio de uma linguagem de programação de alto nível, especificando a estrutura de dados do *software*, os procedimentos e os requisitos envolvido. O resultado desta escrita passa por um processo de conversão das instruções declaradas para uma linguagem de baixo nível, ou linguagem de máquina, para que possa ser interpretado e executado pelo computador ou outro dispositivo.

ENGENHARIA DE *SOFTWARE*

Como o desenvolvimento de *software* envolve uma série de fases e características, para Sommerville (2011), a engenharia de *software* tem como foco abranger todos os aspectos deste desenvolvimento, a partir da especificação até a fase de manutenção do *software*, abrangendo desde processos técnicos, até o gerenciamento de projetos e desenvolvimento de ferramentas. Tem por objetivo a busca de resultados de qualidade, respeitando o tempo e o custo requeridos, sendo utilizado métodos sistemáticos e organizados, pois são mais eficientes na criação de sistemas de alta qualidade.

Câmara, Jandl e Pereira (2014) explana que a qualidade do software é um requisito indispensável para a participação e competitividade no mercado, uma

vez que os clientes esperam rapidez e eficiência na solução de problemas. Neste sentido, a engenharia de software se torna um elemento imprescindível para a obtenção de índices de qualidade sempre elevados e progressivos.

Jamil (2005) aborda que a engenharia de software atualmente visa coadunar os processos de desenvolvimento de software com os princípios dos processos organizacionais, agregando qualidade na gestão empresarial.

Pressman (1995) cita que a engenharia de *software* concilia a engenharia de sistema e de hardware e é formada por um conjunto de três elementos primordiais, que são métodos, ferramentas e procedimentos, os quais fornecem um melhor controle do desenvolvimento e uma base para a criação de produtos de *software* de qualidade superior.

De acordo com as ideias de Azevedo Junior e Campos (2008, p. 28), entende-se que “a engenharia de software se produz através de um conjunto de fases. Cada uma das fases pode envolver métodos, ferramentas e procedimentos, cujas formas de estruturação são citadas como modelo de engenharia de software”. Entende-se que os métodos mencionados, segundo Pressman (1995), indicam uma descrição de “como fazer” durante o desenvolvimento, incluindo as atividades de planejamento, estimativa, análise dos requisitos, codificação e manutenção. Por sua vez, as ferramentas são funções apoiadoras dos métodos, como as ferramentas CASE (*Computer-Aided Software Engineering*) ou engenharia de *software* apoiada por computador. Já os procedimentos realizam a intermediação e ligação entre os métodos e as ferramentas, permitindo que o desenvolvimento seja realizado de maneira racional e oportuna.

Para Sommerville (2011), a seleção de quais técnicas serão utilizadas no desenvolvimento irá depender do tipo de aplicação que se deseja desenvolver, uma vez que existem vários tipos distintos de aplicações.

LEVANTAMENTO DE REQUISITOS

Os requisitos do sistema, de acordo com Sommerville (2011), corresponde àquilo que o cliente necessita como função do *software*, ou seja, é aquilo que o sistema deve fazer, bem como as restrições requeridas envolvidas na solução de um problema. Pfleeger (2004, p. 111) define requisito como “uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar,

para atingir os seus objetivos". Guedes (2011) cita que é nesta fase que se analisa a viabilidade do *software*, verificando se é possível ou não seu desenvolvimento. As atividades que levam ao levantamento, documentação, análise e verificação destas necessidades são conhecidas como engenharia de requisitos.

Pressman (1995) aborda o fato de que, para que o desenvolvimento de *software* possa ser bem-sucedido, é necessária uma completa compreensão dos requisitos levantados, o que se consegue, na maioria das vezes, de acordo com Guedes (2011) por meio de entrevista. Como existem diferentes tipos de pessoas envolvidas no desenvolvimento de *software* profissional, diferentes níveis de abstrações são evidenciados neste cenário. Pressman (1995, p. 231) determina que "a tarefa de análise de requisitos é um processo de descoberta, refinamento, modelagem e especificação". Esta atividade, em primeira análise, pode parecer simples, embora possam surgir ambiguidades que podem prejudicar o desenvolvimento, pois o conteúdo das comunicações é bastante elevado.

Sommerville (2011), aborda os requisitos em dois diferentes níveis de abstração: requisitos de usuário e requisitos de sistema. Os requisitos de usuário são descrições em nível mais alto, em linguagem natural, incluindo diagramas das funções e restrições do sistema. Já os requisitos de sistema, são declarações em nível mais baixo das funções e restrições do *software*, sendo detalhado especificamente tudo que deve ser implementado. Os requisitos em geral, de acordo com Guedes (2011), são classificados frequentemente em dois tipos distintos: requisitos funcionais e requisitos não funcionais devem ser levantados formal e cuidadosamente.

REQUISITOS FUNCIONAIS

Os requisitos funcionais, de acordo com Sommerville (2011), correspondem ao que o sistema irá fazer, descrevem as funções que necessitarão estar contidas no *software*. Dependem de uma série de fatores, como o tipo de sistema requerido, quais serão os futuros utilizadores etc. Guedes (2011) indica que os requisitos funcionais se referem àquilo que os usuários desejam que o sistema realize. Pfleeger (2004) explica que requisitos funcionais representam uma interação do sistema e seu ambiente; todavia, as características da implementação ficam a cargo dos desenvolvedores.

Pfleeger (2004) alerta que os clientes podem não conseguir expressar exatamente suas necessidades, por isso, percebe-se, de acordo com Sommerville (2011), que a especificação dos requisitos funcionais deve ser completa e consistente, com o fim de eliminar ao máximo quaisquer eventuais contratempos e barreiras, registrando todos os serviços que o cliente solicitar e de forma a esclarecer ocorrências de contradições, visto que a engenharia de *software* costuma enfrentar sérios problemas ao se levantar requisitos de forma imprecisa, os quais podem apresentar ambiguidades que podem ser entendidas pelo desenvolvedor de forma diferente daquilo que o cliente realmente solicitou e geralmente ocorre em sistemas complexos e que envolve muitos *stakeholders*. Esta interpretação errônea dos requisitos, por vezes, acarreta em alteração dos requisitos e do sistema, o que refletirá em atraso da entrega e aumento dos custos envolvidos. (SOMMERVILLE 2011).

REQUISITOS NÃO FUNCIONAIS

Os requisitos não-funcionais, embora não sejam funções específicas manipuláveis pelos usuários, estão ligadas a uma ou mais funções do sistema, ou seja, aos requisitos funcionais e, de acordo com Sommerville (2011), são mais críticos que estes. Guedes (2011) cita que eles correspondem a situações que envolvem os requisitos funcionais e, de acordo com Pfleeger (2004), impõe restrições no sistema. Assim, estão relacionados com questões relativas à confiabilidade, tempo de resposta, proteção, desempenho, disponibilidade ou ainda identificação de regras de negócio etc. Caso um requisito não funcional não seja atendido, pode resultar, em alguns casos, na inutilização total do sistema. Tais requisitos, segundo Sommerville (2011), podem surgir em função dos requisitos de produto, organizacionais ou fontes externas.

Requisitos de produto relaciona-se com a determinação ou restrição do comportamento do sistema, abrangendo questões de usabilidade, eficiência, confiança, proteção etc. Requisitos organizacionais provem de políticas e outros fatores próprios da organização do cliente e do desenvolvedor, abrange requisitos ambientais, operacionais, de desenvolvimento etc. As fontes externas abrangem agentes reguladores, leis, fatores éticos e tudo que está externo ao *software* e ao seu desenvolvimento.

LINGUAGEM DE MODELAGEM UNIFICADA – UML

De acordo com Medeiros (2004) e Guedes (2011), a UML (*Unified Modeling Language*) ou Linguagem de Modelagem Unificada, é uma ferramenta destinada a auxiliar os desenvolvedores na modelagem de sistemas, independente da complexidade envolvida, fornecendo um padrão para a elaboração dos planos de arquitetura de projetos de sistemas, seus requisitos, seu comportamento etc. Pfleeger (2004) cita que esta linguagem pode sofrer adaptações para que possa de adequar às várias situações durante o desenvolvimento do *software*.

A UML teve seu surgimento, segundo Guedes (2011), a partir da união de três modelos de modelagem mais utilizados na década de 1990 pelos profissionais da área: Booch, OMT e OOSE, sendo todos eles métodos orientado a objeto. Em 1997, a linguagem passou a ser adotada pela OMG (*Object Management Group*) ou Grupo de Gerenciamento de Objetos como método padrão de modelagem.

Guedes (2011, p. 19) aborda a UML como sendo “uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos”. O mesmo autor ainda cita que a UML é empregada em todo o mundo pela indústria de engenharia de *software*. Na versão 2.0, de acordo com Medeiros (2004), ela passa a ser composta pela descrição dos casos de uso e outros 13 diagramas. Entretanto, esta linguagem não tem a finalidade de apresentar como se deve construir o sistema, mas sim proporcionar formas voltadas à representação do mesmo em seus diferentes estágios durante o desenvolvimento. Portanto, ainda de acordo com Medeiros (2004), a UML se caracteriza como uma forma de comunicação que um processo pode adotar, oferecendo, como cita Guedes (2011), uma grande quantidade de diagramas que abordam características estruturais e comportamentais de um *software*.

Para Guedes (2011), todo sistema precisa de ser modelado previamente, uma vez que com o passar do tempo o mesmo costuma evoluir, seja para adequação às novas necessidades dos clientes, seja para acompanhar o mercado, ou ainda para se adequar à nova legislação. Estas ocorrências acabam implicando no aumento da complexidade e abrangência do *software*.

DIAGRAMA DE CASO DE USO

Ao se utilizar a UML em projetos orientados a objetos, o diagrama de caso de uso acaba se tornando a parte mais importante do desenvolvimento, segundo Medeiros (2004), pois este está presente em todos os estágios de desenvolvimento do *software*, funcionando como uma ferramenta de consulta, acerto, discussão, reuniões e alterações. Guedes (2011, p. 52) aponta que este diagrama objetiva "apresentar uma visão externa geral das funcionalidades que o sistema deverá oferecer", embora ele não se preocupe em como as funções serão implementadas, tratando-se de um diagrama que pode ser utilizado por todos os outros diagramas como referência. Medeiros (2004, p. 36) afirma que "ele é a análise intrínseca de um negócio, dentro do processo de desenvolvimento de *software*, sugerido pelo modelo iterativo e por outras metodologias que utilizam a UML". Enfim, em uma visão mais generalista, um caso de uso é uma representação escrita de uma ou várias ações.

Guedes (2011) apresenta que este diagrama emprega dois itens principais: atores e caso de uso. Os atores são apresentados graficamente como bonecos magros e representam os usuários que farão uso das funções e serviços do sistema, sendo que o ator pode representar uma pessoa, um hardware ou outro sistema que realize alguma interação com o sistema. Sintetizando, um ator se refere a qualquer entidade externa que interaja com o *software*, incluindo, abaixo da sua representação gráfica, seu papel assumido no diagrama. Os casos de uso são apresentados graficamente como elipses contendo no seu centro uma breve descrição da sua funcionalidade e representam os requisitos levantados, ou seja, as funções e serviços do sistema que serão manipulados pelos atores. Pfleeger (2011) apresenta, além dos atores e casos de uso, a extensão, que tem a função de ampliar um caso de uso com o fim de abordá-lo sob uma outra perspectiva ou possibilitar uma análise mais profunda.

Casos de uso precisam de um nível de detalhe relativamente grande e objetivo, uma vez que ele será utilizado durante todo o desenvolvimento, por todas as outras partes. Guedes (2011) cita que é necessário que se empregue uma linguagem mais simples, que facilite sua compreensão, o que irá permitir que os usuários possam absorver a forma como o sistema irá se comportar.

DIAGRAMA DE CLASSE

O diagrama de classe, como comenta Medeiros (2004) não foi criado pela UML, sendo uma invenção de diversos autores, o que resultou em diferentes formas de sua criação. Com o surgimento da UML, as melhores práticas de criação deste diagrama foram adotadas pela ferramenta e as demais formas deixaram de ser utilizadas, o que contribuiu para a criação de uma padronização para que os desenvolvedores pudessem seguir para facilitar a criação do diagrama.

Como a UML é destinada à modelagem de sistemas que empregam o paradigma de programação orientadas a objeto, como afirma Guedes (2011), e este paradigma, conforme apresenta Deitel e Deitel (2010), se concentra basicamente nos conceitos de classes e objetos, nos quais se criam representações do mundo real. Como exposto por Guedes (2011), sua utilização tem como objetivo a apresentação das classes que serão implementadas no sistema, detalhando seus métodos e atributos, e ainda como se dá o relacionamento e a comunicação entre elas. O autor ainda exalta sua grande importância devido ao fato de que, assim como o diagrama de caso de uso, ele é empregado na criação da maioria dos demais diagramas utilizados.

As classes do diagrama, como afirma Medeiros (2004), são compostas por atributos e métodos. Os atributos representam as características da classe e, para Guedes (2011), eles comportam os detalhes que variam de um objeto para outro. Objetos, ou instâncias, são exemplos das classes, como se uma classe representasse uma categoria e o objeto seus membros. Os métodos, de acordo com Medeiros (2004) são a funções da classe. É aquilo que o objeto da classe realizará durante a execução do sistema. Guedes (2011) conceitua os métodos como uma operação que uma instância poderá realizar ao executar um conjunto de instruções. Para que as classes possam interagir entre si e trocar informações, são empregadas as associações, também conhecida como relacionamento, entre uma ou mais classes do sistema, como afirma Medeiros (2004). De acordo com Guedes (2011), este relacionamento é representado por linhas interligando as classes envolvidas e, para facilitar sua compreensão, podem ter título ou nome.

DIAGRAMA DE ENTIDADE E RELACIONAMENTO

O Diagrama de Entidade e Relacionamento (DER) tem como foco a representação gráfica do Modelo de Entidade e Relacionamento (MER), como explica Rodrigues (2015). O Modelo de Entidade e Relacionamento foi desenvolvido em 1976 por Peter Chen e é referência para a criação do modelo conceitual de banco de dados e seu emprego visa a descrição das entidades do negócio, suas características, bem como suas associações, ou relacionamentos, tomando como base características do mundo real, de acordo com Heuser (1998). Sintetizando, o Diagrama de Entidade e Relacionamento representa uma abstração da estrutura em que será construído o banco de dados da aplicação, tendo como elementos fundamentais as entidades e os relacionamentos. Rodrigues (2015) cita que este diagrama facilita a comunicação entre a equipe, uma vez que “oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado”

As entidades constituem o conceito principal da abordagem Entidade-Relacionamento, conforme Heuser (1998), e devem ser nomeadas com substantivos que remetam claramente à sua função. Representam objetos do mundo real sobre a qual se quer manter algum registro e podem ser físicas ou lógicas. No Diagrama de Entidade e Relacionamento as entidades são representadas por um retângulo. As entidades físicas são aquelas existentes e tangíveis no mundo real, como um cliente ou um produto. As entidades lógicas não são existentes fisicamente no mundo real, sendo fruto da interação entre as entidades do negócio, como uma venda por exemplo, que se consolida a partir da ação exercida da entidade cliente sobre a entidade produto (RODRIGUES, 2015).

Os relacionamentos no Diagrama de Entidade e Relacionamento, nos dizeres de Heuser (1998), são representados por losango que se ligam às entidades. Rodrigues (2015) cita que estes devem ser nomeados com verbos que indiquem o teor da interação entre as entidades vinculadas. Para Rodrigues (2015), as associações indicam como se dá o relacionamento entre as entidades. Uma característica importante presente na associação, segundo Heuser (1998), são as cardinalidades, que indicam a quantidade mínima e máxima de ocorrências de uma entidade sobre a outra. Estas cardinalidades, segundo Rodrigues (2015) podem

ser: 1..1 (lê-se um para um), 1..n (lê-se um para muitos), ou n..n (lê-se muitos para muitos).

O relacionamento 1..1 indica que, em um relacionamento entre duas entidades, cada uma delas faz referência a apenas uma unidade da outra. No relacionamento 1..n, tem-se que uma entidade de um relacionamento pode referenciar várias instâncias da outra, enquanto que, em sentido contrário, só pode haver uma referência das instâncias referenciadas registrada na outra entidade. No relacionamento n..n, cada entidade pode referenciar várias unidades da outra, de ambos os lados. (RODRIGUES, 2015).

Por fim, Rodrigues (2015) apresenta os atributos, os quais são destinados a descrever as características de cada entidade do domínio. Para Heuser (1998, p. 25), atributo é um "dado que é associado a cada ocorrência de uma entidade ou de um relacionamento". Neste sentido, cada instância de uma entidade guardará informações que serão relacionadas a uma instancia de uma outra entidade do domínio.

LINGUAGEM DE PROGRAMAÇÃO JAVA

A linguagem de programação Java é uma linguagem de alto nível orientada a objetos. Teve seu surgimento no ano de 1991, chamada inicialmente de Oak, e tinha como foco a convergência entre diversos dispositivos eletrônicos. No ano de 1995, com o propósito de adaptar a linguagem Oak para a internet, seu criador, James Gosling, criou a plataforma Java, que, segundo Luckow e Melo (2015), tinha como diferencial em relação outras linguagens da época a premissa de ser executada sobre a Máquina Virtual Java – JVM, que permitiu a adaptação multiplataforma da linguagem. Neste sentido, todos os dispositivos que possam executar a máquina virtual serão capazes de trabalhar com a linguagem Java, validando seu slogan: *"write once, run anywhere"*, que traduzido fica: "escreva uma vez, execute em qualquer lugar".

Segundo Deitel e Deitel (2010), os programas desenvolvidos nesta linguagem passam por 5 fases diferentes: edição, compilação, carregamento, verificação e execução. Desde o seu surgimento, a linguagem passou por diversas implementações em busca de melhorias para melhor se adaptar as mudanças ocorridas no mundo. O meio em que os usuários interagem com as tecnologias da

informação e comunicação foi submetido a grandes mudanças e, dessa forma, para Gonçalves (2007), a linguagem Java, objetivando acompanhar as mudanças, foi adaptada para as mais diversas plataformas, como aplicativos desktops, páginas *web*, aplicações móveis, entre outras.

Hoje, por apresentar grande segurança, grandes bancos fazem uso desta linguagem. Segundo Gonçalves (2007), por possibilitar estabilidade e portabilidade entre empresas e possibilitando grande tráfego de dados, muitas organizações adotaram a linguagem Java para desenvolvimento de suas aplicações. Alguns exemplos de utilizadores da linguagem são: NASA, IBM, ESPN entre outros, devido à sua grande confiabilidade apresentada. Nesta ótica, para Deitel e Deitel (2010), ela é utilizada, como exemplo, por servidores web, banco de dados relacionais, sistemas de cartões de crédito etc.

BANCO DE DADOS

Para que uma aplicação possa atuar de forma a satisfazer seus propósitos, é necessário que os dados trabalhados possam ser guardados propiciando uma consulta posterior. Segundo Setzer e Silva (2005) para satisfazer esta necessidade foi criado o banco de dados, uma tecnologia destinada a funcionar como um repositório de dados, o que possibilita que diversas aplicações possam atuar sobre uma mesma base.

Abreu e Machado (2012) cita que que o fundamento básico do banco de dados é a definição e manutenção dos dados, independente de qual sistema o utilize. Dessa forma, ao passo que os dados são consultados e relacionados entre si possibilitando a geração de um significado dentro de um contexto, estes acabam se tornado informação aos seus usuários, refletindo características de interesse do mundo real e que deve ser encarada como um bem patrimonial da organização assim como os recursos humanos, materiais, financeiros etc.

Segundo Heuser (1998) a construção de um projeto de banco de dados normalmente se dá em três etapas: modelagem conceitual, lógica e física. Na modelagem conceitual, procura-se projetar a base de dados identificando os modelos de dados abstratos que podem vir a ser armazenados, a fim de definir a sua estrutura. Na modelagem lógica, pretende-se definir as estruturas de dados, a nível de Sistema de Gerenciamento de Banco de Dados, identificadas na

modelagem conceitual. Heuser (1998) cita que a modelagem física é responsável por definir os parâmetros físicos necessários para acesso ao banco de dados, bem como para melhorar a performance do sistema.

BANCO DE DADOS MYSQL

O MySQL, projeto iniciado em 1980, é um Sistema de Gerenciamento de Banco de Dados protegido por licença de *software* livre GPL, o que permite sua utilização grátis caso o aplicativo que o utilize não seja comercializado e, de acordo com Luckow e Melo (2015) é um dos bancos de dados mais utilizados no mundo com mais de 70 milhões de instalações. Sua interface de comunicação é a linguagem SQL. Pacievitch (2011) cita alguns dos principais motivos de sua grande utilização e seu constante crescimento: facilidade em que os usuários encontram em sua manipulação e ainda por permitir sua execução em diversos sistemas operacionais e ainda por ser *software* livre. Neste sentido, grandes organizações adotam esta tecnologia de banco de dados, tais como NASA, HP, Bradesco, Sony, entre outras.

Comparando-se sua utilização frente a outros bancos de dados também utilizados atualmente, de acordo com as ideias de Pacievitch (2011), é possível perceber algumas vantagens, como uma facilidade maior para programação, disponibilização de funções mais simplificadas e ainda a possibilidade de modificação, permitindo uma maior adaptação as necessidades dos usuários, contanto que estes tenham boas habilidades na arte da programação.

JASPERREPORTS E IREPORT

Para que um sistema, seja ele simples ou complexo, possa fornecer informações que auxiliarão na tomada de decisão, segundo Lanhellas (2014), é de grande importância que o mesmo seja capaz de gerar relatórios das informações manipuladas pelos usuários durante suas atividades. Estes relatórios, segundo Alexandre (2012), permitirão a apresentação de informações de forma resumida ou detalhada, dependendo da necessidade, e organizada, favorecendo a tomada de decisão correta.

Para Luckow e Melo (2015), os *frameworks* JasperReports e Ireport são

ferramentas muito importantes no desenvolvimento em Java, pois permite, respectivamente, que um relatório possa ser gerado nos mais diversos formatos, como PDF, XML, XLS etc. e possibilita que um relatório possa ser criado a partir de um ambiente gráfico, dispensando a manipulação direta de arquivo XML, o que fica a cargo do *framework*. Alexandre (2012) cita que o JasperReports oferece ao desenvolvedor um leque de funcionalidades capaz de criar relatórios complexos e estruturados, a partir do *template* criado pelo Ireport, que é um arquivo XML com a extensão ".jrxml".

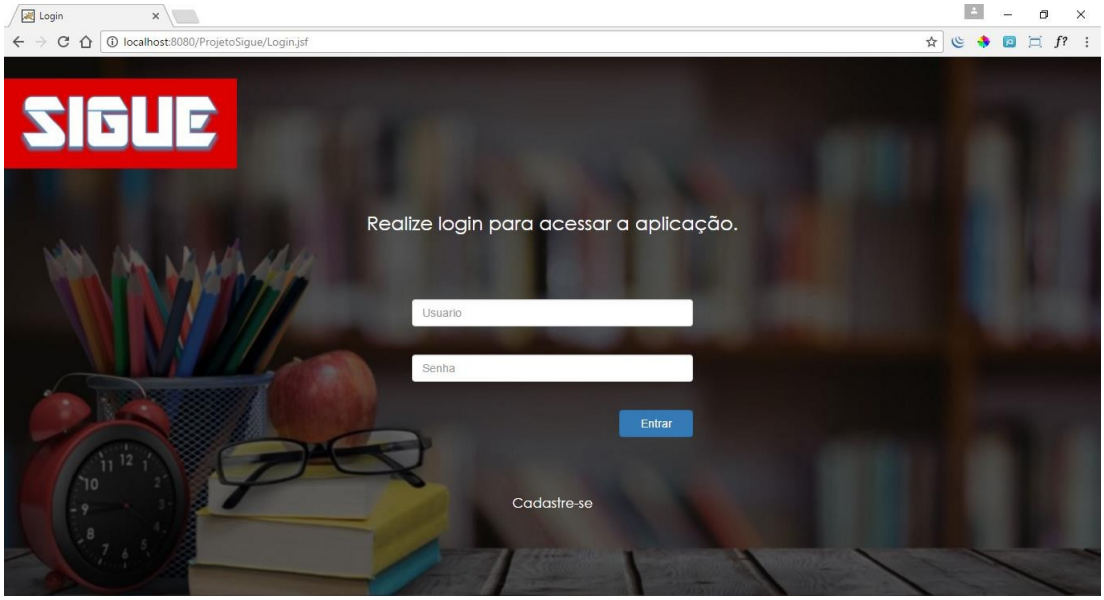
Lopes (2012) conceitua o Ireport como sendo uma IDE que permite o desenvolvimento do Layout do relatório, ou aquilo que será visto graficamente. Ainda no mesmo sentido, Lanhellas (2014) expõe que o Ireport permite "desenhar" o relatório, assim como se necessita que as informações sejam mostradas. De acordo com Alexandre (2012), o JasperReports é um *framework* Java capaz de gerar relatórios dinâmicos para *softwares* Java, de forma fácil e exigindo poucas linhas de código a serem implementadas.

A partir destas informações, o *software* SIGUE, que está sendo desenvolvido juntamente com este trabalho, contará com as tecnologias JasperReports e Ireport para a criação dos seus relatórios, a fim de apresentar informações uteis para os usuários do sistema, objetivando auxiliar na tomada de decisão dos gestores escolar.

APRESENTAÇÃO DO SISTEMA

Tendo como referência o levantamento bibliográfico apresentado anteriormente, um *software* deve empregar as melhores práticas de mercado, para assim poder gerar uma informação de qualidade para as organizações. Nesse sentido, no intuito de auxiliar uma organização educacional, apresenta-se o presente sistema denominado SIGUE. A utilização do sistema desenvolvido com este trabalho se dá a partir da autenticação do usuário na tela de login, como ilustrada na figura 1.

FIGURA 1 - Tela de login



Fonte: Desenvolvida pelo autor.

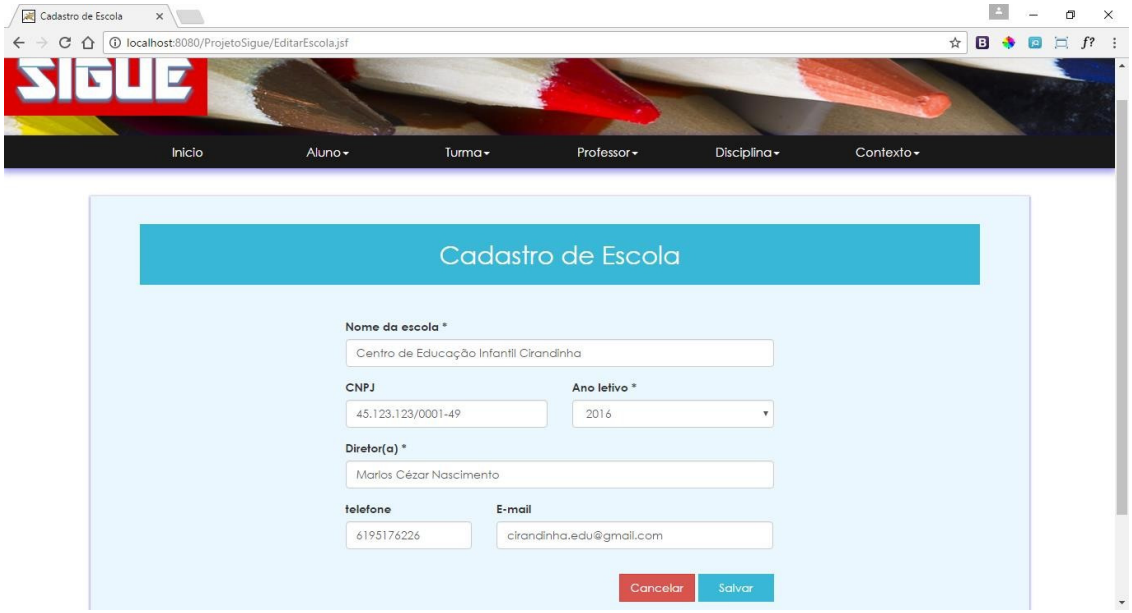
Caso o usuário não possua um cadastro no sistema, ele poderá clicar no link cadastre-se da tela de login, como na figura 1. Dessa forma ele será redirecionado para a pagina de cadastro de usuário, como ilustrada na figura 2:

FIGURA 2 - Tela de cadastro de usuário

Fonte: Elaborada pelo autor.

Para que a aplicação funcione corretamente, é necessário que que a escola seja cadastrada. Este cadastro pode ser realizado a partir da tela de cadastro de escola, conforme ilustrada na figura 3.

FIGURA 3 – Tela de cadastro de escola.



Fonte: Elaborada pelo autor.

Após a autenticação do usuário, o mesmo será redirecionado para a

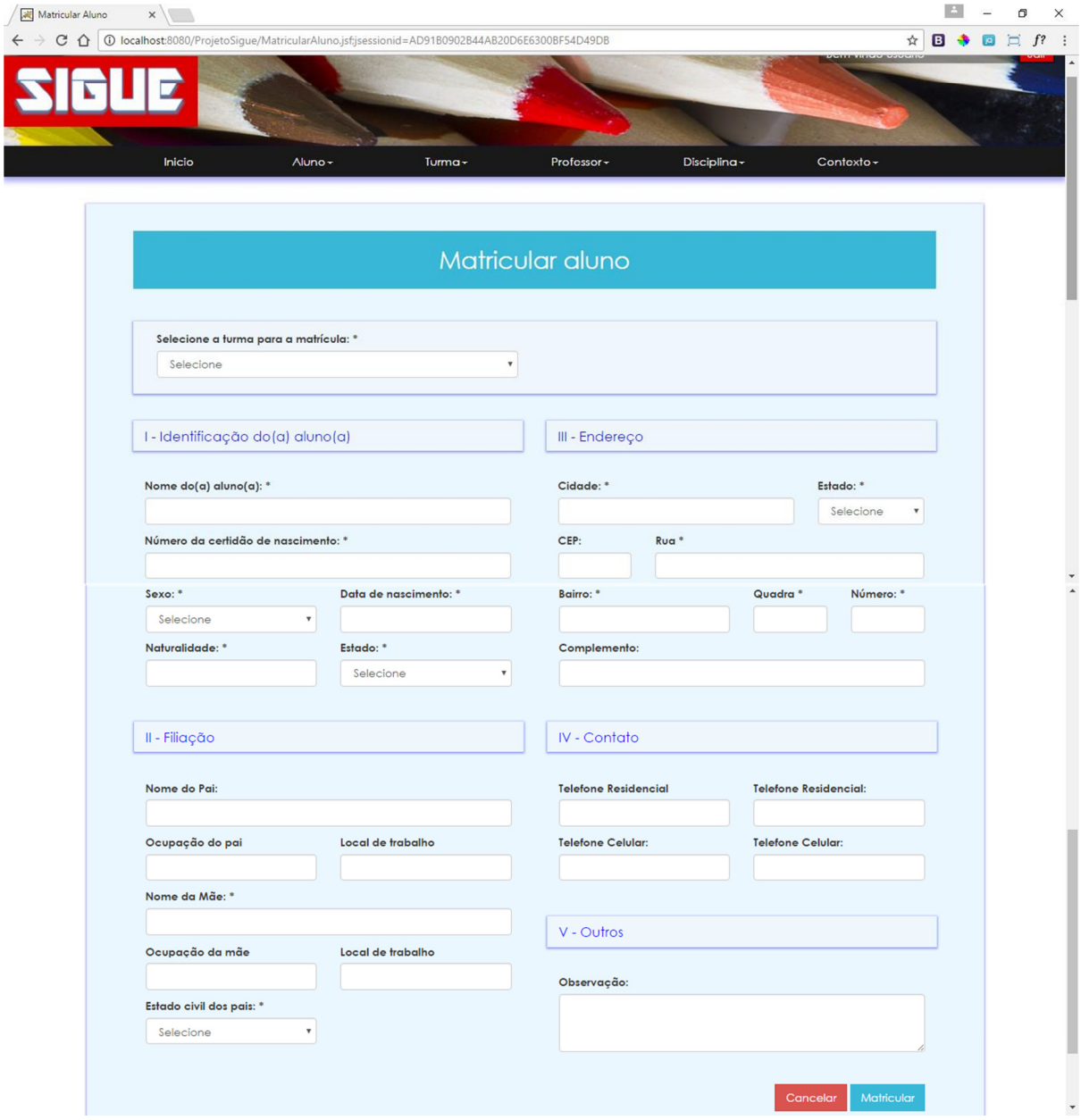
página inicial do sistema, como ilustrada na figura 3. Nesta tela, o usuário poderá visualizar os principais módulos do sistema incluídos na barra de menu, os quais foram desenvolvidos a partir dos requisitos levantados e que podem ser visualizados por meio do diagrama de caso de uso incluído no apêndice A. Além disso, no centro da tela, estarão expressas algumas informações resumidas a respeito da escola cadastrada.

FIGURA 4 – Tela inicial do sistema

Fonte: Desenvolvida pelo autor.

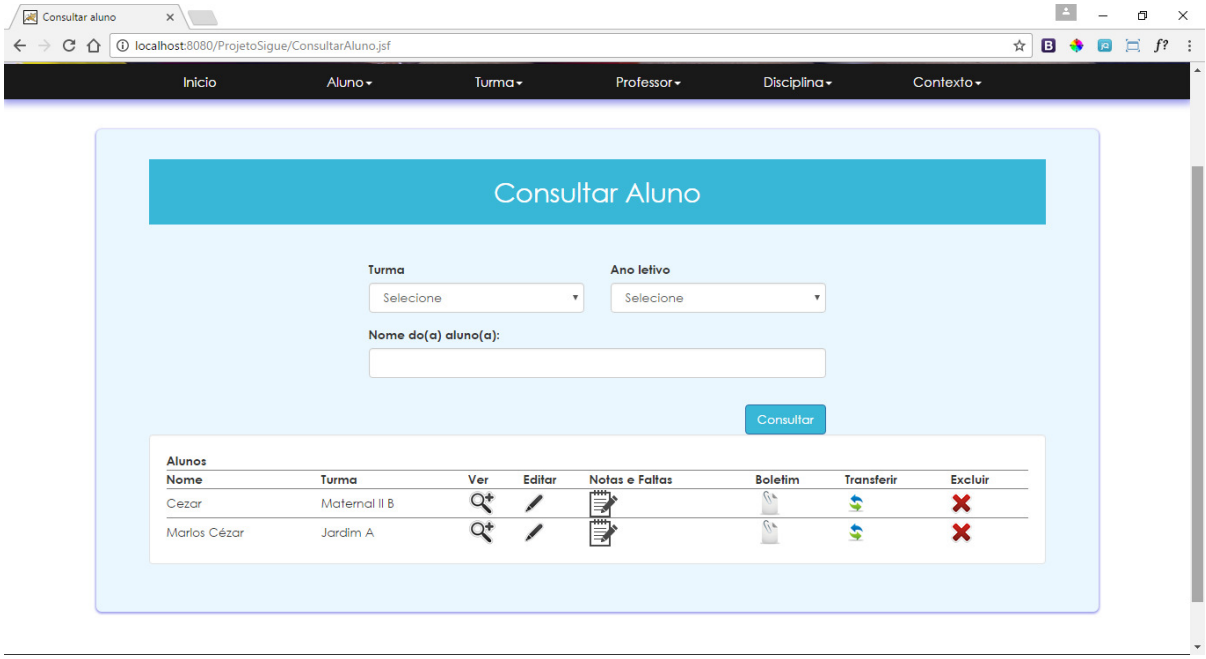
A partir do diagrama de caso de uso citado acima, foi desenvolvido o diagrama de classes, o qual foi incluído no apêndice B. Tal diagrama de classe expressa de maneira bem clara como será a interação entre as classes do sistema, respeitando as características expressas no levantamento de requisitos. Neste sentido, ao se visualizar o diagrama, logo se perceberá que um aluno deve ser incluído em uma turma. Este cenário será concretizado a partir da matricula do aluno, por meio da tela representada na figura 5, sendo esta uma das principais funções do *software*. Na figura 6, pode-se observar a tela de consulta de aluno, a qual pode ser realizada por turma e por ano letivo.

FIGURA 5 – Tela de matrícula de aluno.



Fonte: Elaborada pelo autor.

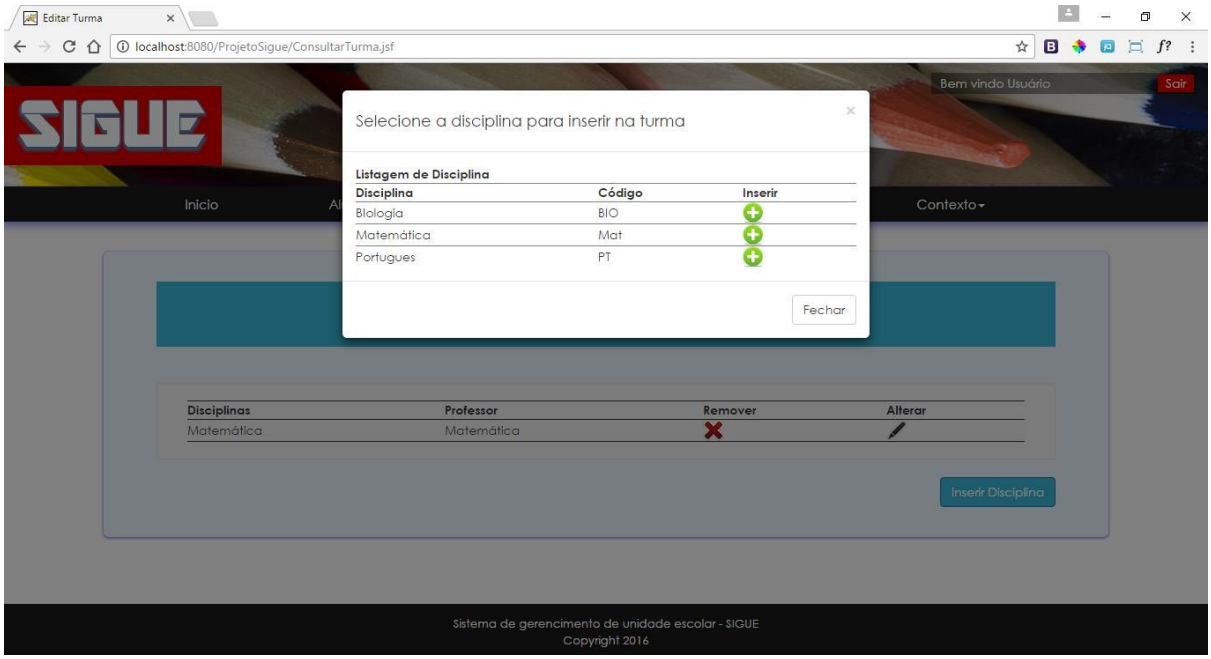
FIGURA 6 – Tela de consulta de aluno.



Fonte: Elaborada pelo autor.

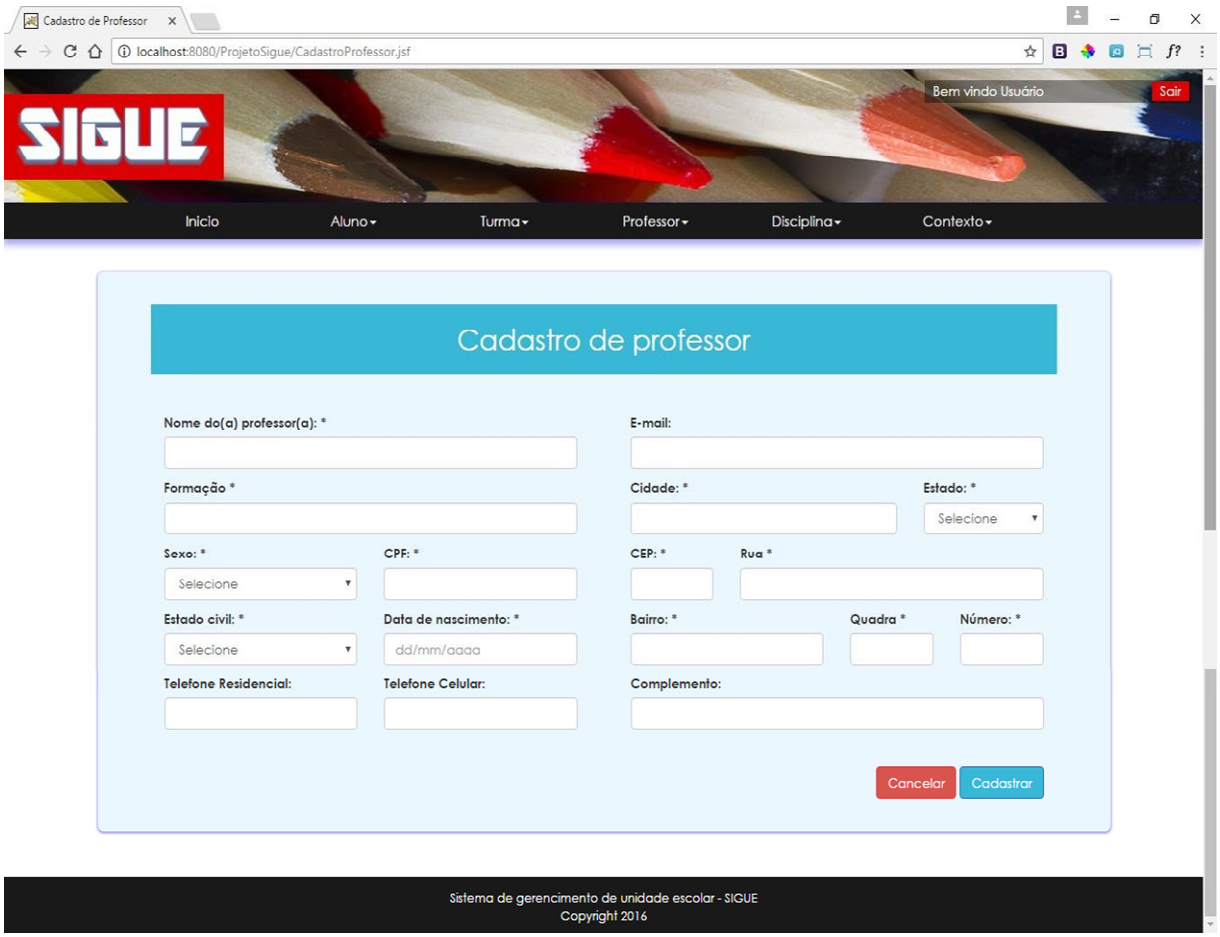
Outra função bem importante do *software* é a gestão das turmas. Neste ponto, após o cadastro da turma, deverão ser inseridas suas disciplinas, como ilustrado na figura 7. As disciplinas em cada turma têm seus respectivos professores, dessa maneira é preciso cadastrá-los no sistema, como ilustrado na figura 8, para que eles possam ser vinculados a uma disciplina.

FIGURA 7 – Tela de inserção de disciplina na turma.



Fonte: Elaborada pelo autor.

FIGURA 8 – Tela de cadastro de professor.



Fonte: Elaborada pelo autor.

CONCLUSÃO

Conclui-se que o presente trabalho, realizado com o intuito de apresentar uma ferramenta computacional para ser empregada nas escolas de ensino básico, pode contribuir para a agilização e melhoria das atividades administrativas. Nesse sentido, evidencia-se que, à medida que as escolas passam a fazer uso da tecnologia em suas tarefas diárias, em especial um *software* de gerenciamento escolar, suas atividades administrativas ficam possibilitadas de serem executadas de forma a aproveitar melhor o tempo em cada uma delas, bem como na facilitação das mesmas, evitando a produção da grande quantidade de papeis no dia a dia e ainda melhorando o processo de consulta de informações. Toda esta verificação, condiz com as hipóteses suscitadas do problema, confirmando-as; uma vez que a tecnologia da informação e comunicação veio para facilitar o dia a dia das pessoas e das organizações, além de outros aspectos mais específicos de tantas outras abordagens que esta pode ser condicionada que vai além dos limites desta pesquisa. Neste sentido, os objetivos da pesquisa foram atendidos, assim como o problema ficou respondido.

REFERÊNCIAS

ABREU, M. P.; MACHADO, F. N. R. **Projeto de banco de dados**: uma visão prática. São Paulo: Érica, 2012. 320 p.

ALBERTIN, A. L.; ALBERTIN, R. M. de M. Tecnologia de informação e desempenho empresarial no gerenciamento de seus projetos: um estudo de caso de uma indústria. **Revista de Administração Contemporânea**. Curitiba, v.12, n.3, p.599- 629, jul./set. 2008.

ALEXANDRE, M. Gerando Relatórios com JasperReports**. Devmedia**. 2012. Disponível em: <www.devmedia.com.br/gerando-relatorios-com-jasperreports/24798>. Acesso em: 07 jan. 2017.

ANDRADE, G. K.; AUDY, J.L.N.; CIDRAL, A. **Fundamentos de sistemas de informação**. Porto Alegre: Bookman, 2005. 208 p.

AZEVEDO JUNIOR, D. P. de; CAMPOS, R. Definição de requisitos de software baseada numa arquitetura de modelagem de negócios. **Produção**. v. 18, n. 1, p. 26- 46, 2008.

BIO, S. R. **Sistemas de informação**: um enfoque gerencial. 2. ed. São Paulo: Atlas, 2008. 235 p.

CÂMARA, C. E.; JANDL, P. J.; PEREIRA, C. C. A importância da qualidade no desenvolvimento de software. **Revista Engenho**. v. 9, 2014.

CARVALHO, M. M.; LAURINDO, F. J. B.; RABECHINI Jr, R.; SHIMIZU, T**.** O papel da tecnologia da informação na estratégia das organizações. **Gestão e Produção**, São Paulo, v.8, n.2, p.160-179, ago. 2001.

CUKIERMAN, H. L.; PRIKLADNICKI, R.; TEIXEIRA, C. Um olhar sociotécnico sobre a engenharia de software. **Revista de Informática Teórica e Aplicada**. Porto Alegre, v. 14, n. 2, 2007.

DEITEL, H.; DEITEL, P. **Java**: como programar. 8. ed. São Paulo: Person Prentice Hall, 2010. 1114 p.

DUARTE, V. M. do N. Pesquisas: exploratória, descritiva e explicativa. **Monografias Brasil Escola**. 2012. Disponível em: <http://monografias.brasilescola.com/regras/ pesquisas-exploratoria-descritiva-explicativa.htm>. Acesso em: 30 mai. 2016.

ESCRIVÃO FILHO, E.; MORAES, G. D. de A.; TERENCE, A. C. F. A tecnologia da informação como suporte à gestão estratégica da informação na pequena empresa. **Revista de Gestão da Tecnologia e Sistemas de Informação**. São Paulo, v.1, n.1, p.28-44, 2004.

FERREIRA, D. A. L.; MALCHER, P R.C.; OLIVEIRA, S. R. B.; VASCONCELOS, A.
M. L. de. Um mapeamento sistemático sobre abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software. **Revista de Sistemas de Informação da FSMA**. n. 16, p. 3-15, 2015.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2010. 184 p.

GONÇALVES, E. **Desenvolvendo aplicações web com jsp, servlets, javaServer faces, hibernate, ejb 3 persistence e ajax**. Rio de Janeiro: Ciência Moderna, 2007. 776 p.

GUEDES, G. T. A. **UML 2**: uma abordagem prática. 2. ed. São Paulo: Novatec, 2011. 484 p.

HEUSER, C. A. **Projeto de banco de dados**. 4. ed. Porto Alegre: Sagra Luzzato. 1998. 206 p.

JAMIL, G. L. Perspectiva de colaboração da engenharia de software para a gestão da informação e do conhecimento. **Pretexto**. Belo Horizonte, v. 6, n. 2, p. 43-56, 2005.

LAKATOS, E. M.; MARCONI, M. A. **Técnicas de pesquisa**. 6. ed. São Paulo:

Atlas, 2006. 289 p.

LANHELLAS, R. JasperReport: Relatórios em Java com iReport. **Devmedia**. 2014. Disponível em: <www.devmedia.com.br/jasperreport-relatorios-em-java-com-ireport/3 1075>. Acesso em: 07 jan. 2017.

LOPES, C. Como gerar relatórios com Ireport, JasperReport e com Hibernate. **iMasters**. 2012. Disponível em: <imasters.com.br/artigo/23764/java/como-gerar-relatorios-com-ireport-jasperreport-e-com-hibernate/?trace=1519021197&source=sin gle>. Acesso em: 07 jan. 2017.

LUCKOW, D. H.; MELO, A. A. **Programação Java para a web**. 2. ed. São Paulo: Novatec, 2015. 677 p.

MEDEIROS, E. S. **Desenvolvendo software com UML 2.0**. São Paulo: Pearson Makron Books, 2004. 264 p.

MORALES, A. B. T.; ROSSETTI, A. G. O papel da tecnologia da informação na gestão do conhecimento. **Ciência da Informação**, Brasília, v. 36, n. 1, p. 124-135, jan./abr. 2007.

PACIEVITCH, Y. MySQL. **Infoescola**. 2011. Disponível em:<http://www.infoescola.com/informatica/mysql/>. Acesso em: 31 mai. 2016.

PFLEEGER, S. L. **Engenharia de software**: teoria e prática. 2. ed. São Paulo: Prentice Hall, 2004. 537 p.

POTTER, R. E; RAINER JR, R. K; TURBAN, E. **Administração de Tecnologia da Informação**: Teoria e Prática. Rio de Janeiro: Elsevier, 2005.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Pearson Makron Books, 1995. 1056 p.

RODRIGUES, J. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-relacionamento (DER). **Devmedia**. 2015. Disponível em: <www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/1433 2>. Acesso em: 11 nov. 2016.

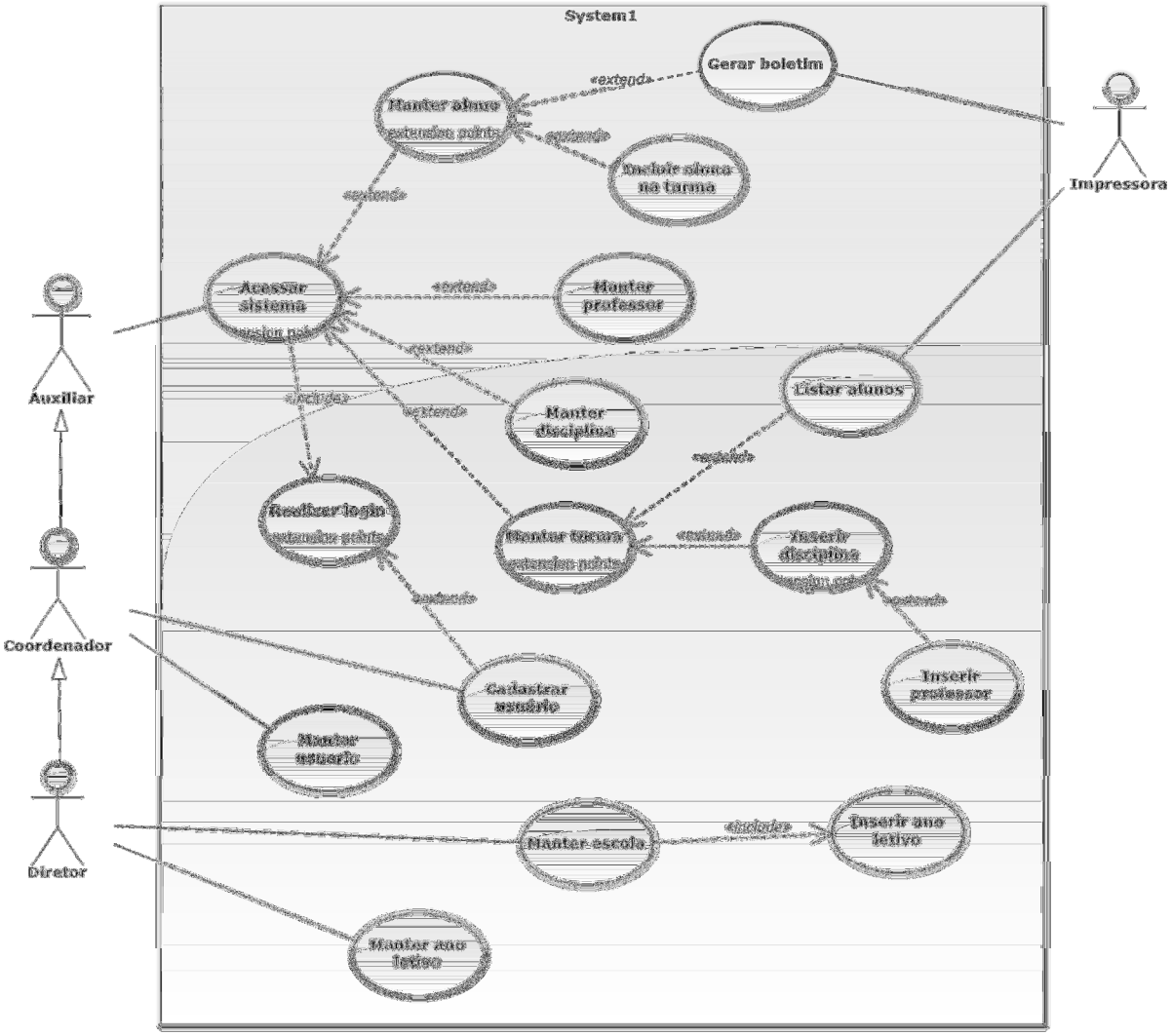
SACILOTTI, A. C. **A importância da tecnologia da informação nas micro e pequenas empresas**: um estudo exploratório na região de Jundiaí. 2011, 116 f. Dissertação (Mestrado em Administração) – Faculdade Campo Limpo Paulista, São Paulo, 2011.

SETZER, V.W.; SILVA, F.S.C. **Banco de dados**: aprenda o que são, melhore seu conhecimento, construa os seus. São Paulo: Edgard Blucher, 2005. 380 p.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.

APÊNDICE A – Diagrama de caso de uso

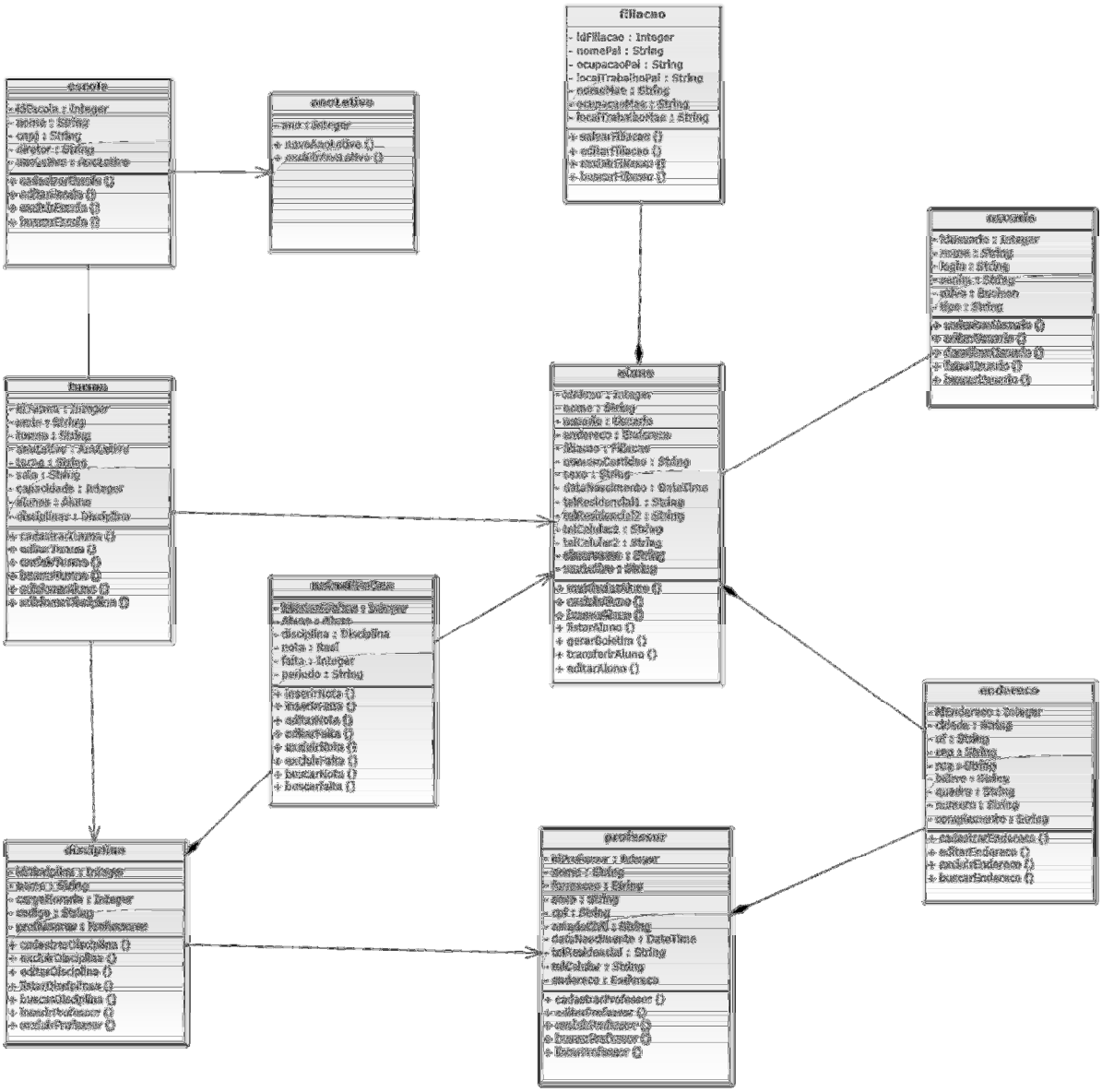
FIGURA 9 – Diagrama de caso de uso.



Fonte: Elaborada pelo autor.

APÊNDICE B – Diagrama de classe

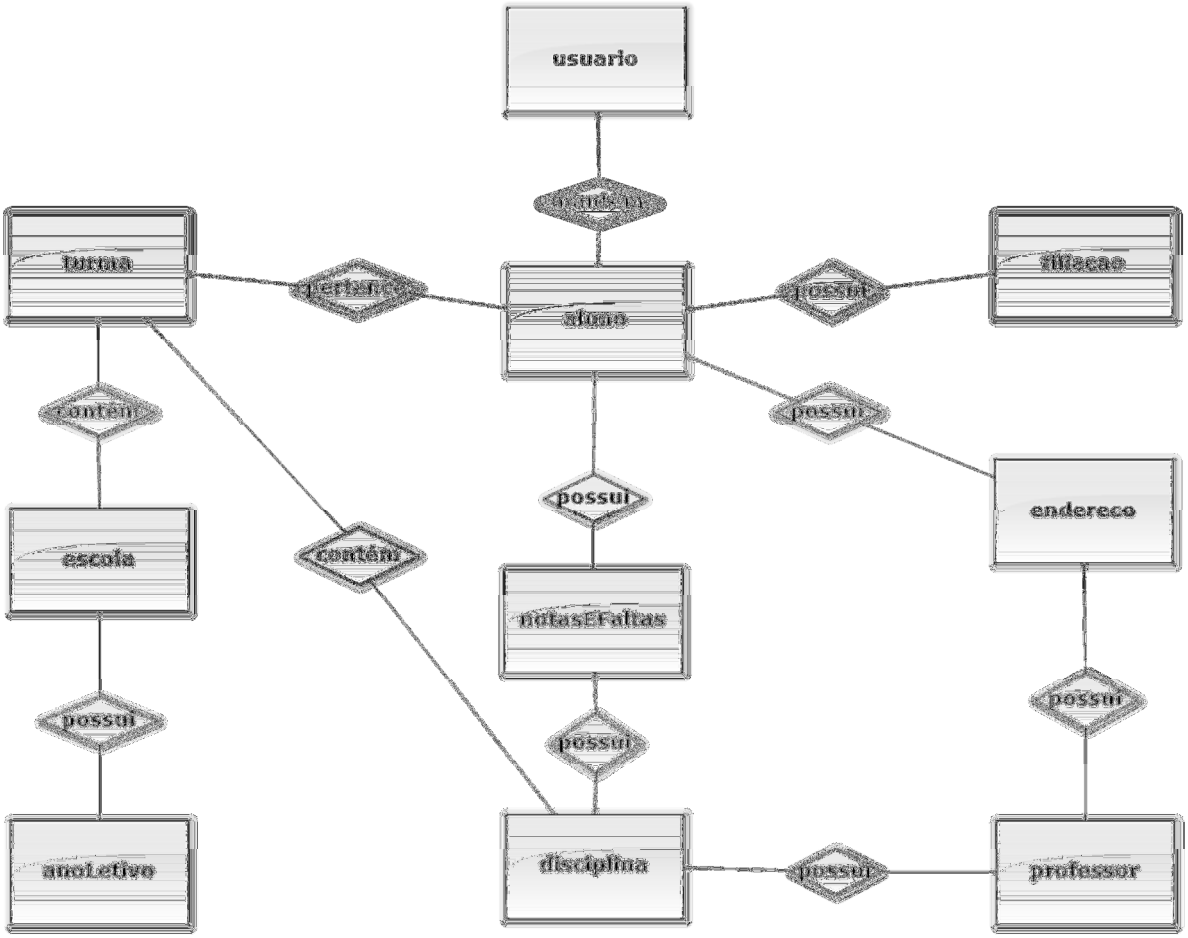
FIGURA 10 – Diagrama de classe



Fonte: Elaborada pelo autor.

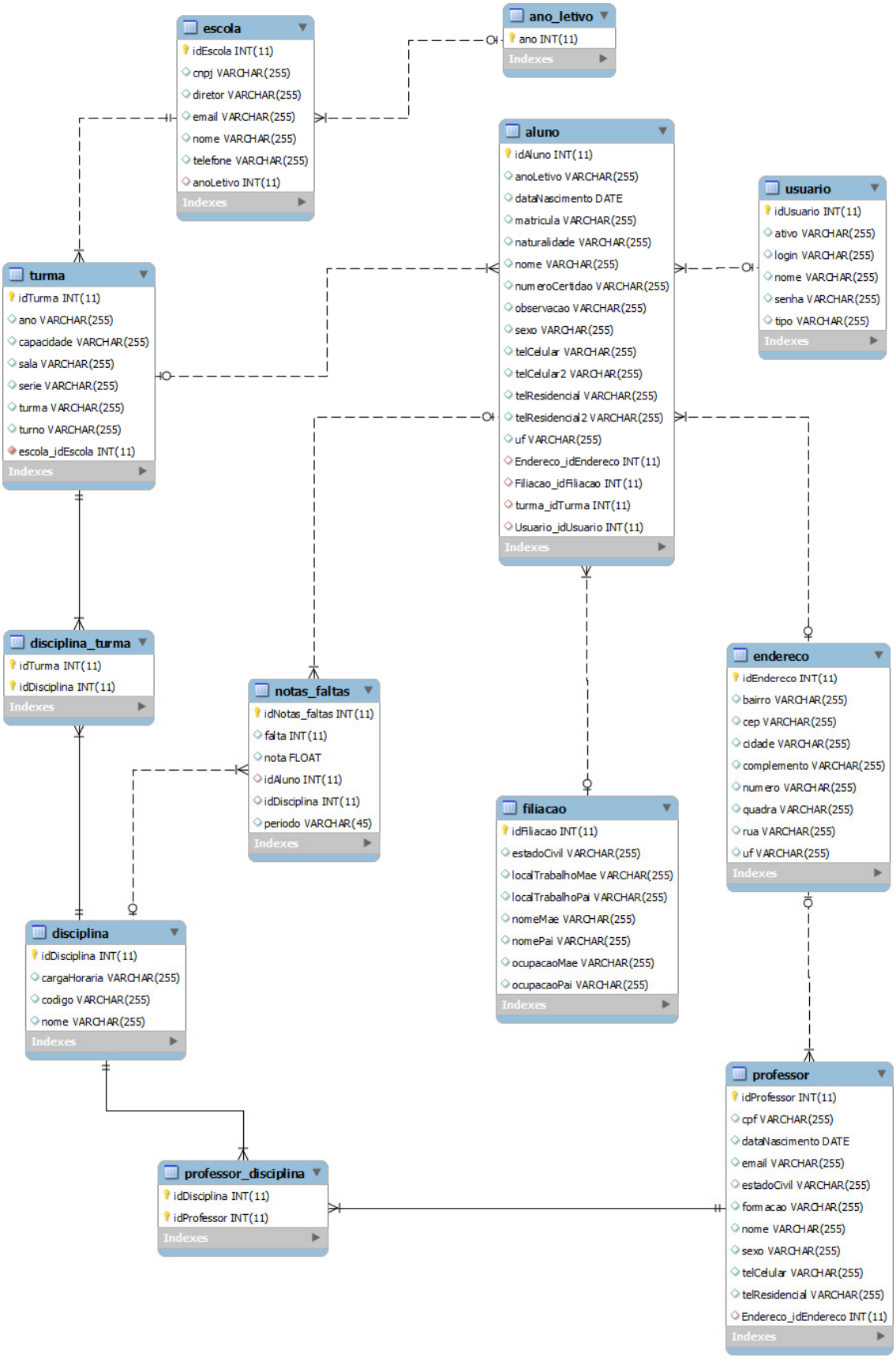
APÊNDICE C – Diagramas de entidade e relacionamento

FIGURA 11 – Diagrama conceitual de entidade e relacionamento



Fonte: Elaborada pelo autor.

FIGURA 12 – Diagrama lógico de entidade e relacionamento



Fonte: Elaborada pelo autor.

APÊNDICE D – Lista de requisitos

QUADRO 1 – Lista de requisitos funcionais

Cód.	Requisito
RF-01	Manter usuários
RF-02	Manter aluno
RF-03	Manter turma
RF-04	Manter disciplina
RF-05	Manter escola
RF-06	Manter ano letivo
RF-07	Manter notas de aluno
RF-08	Gerar boletim
RF-09	Adicionar disciplinas na turma
RF-10	Adicionar professor na disciplina
RF-11	Adicionar aluno na turma

Fonte: Elaborado pelo autor.

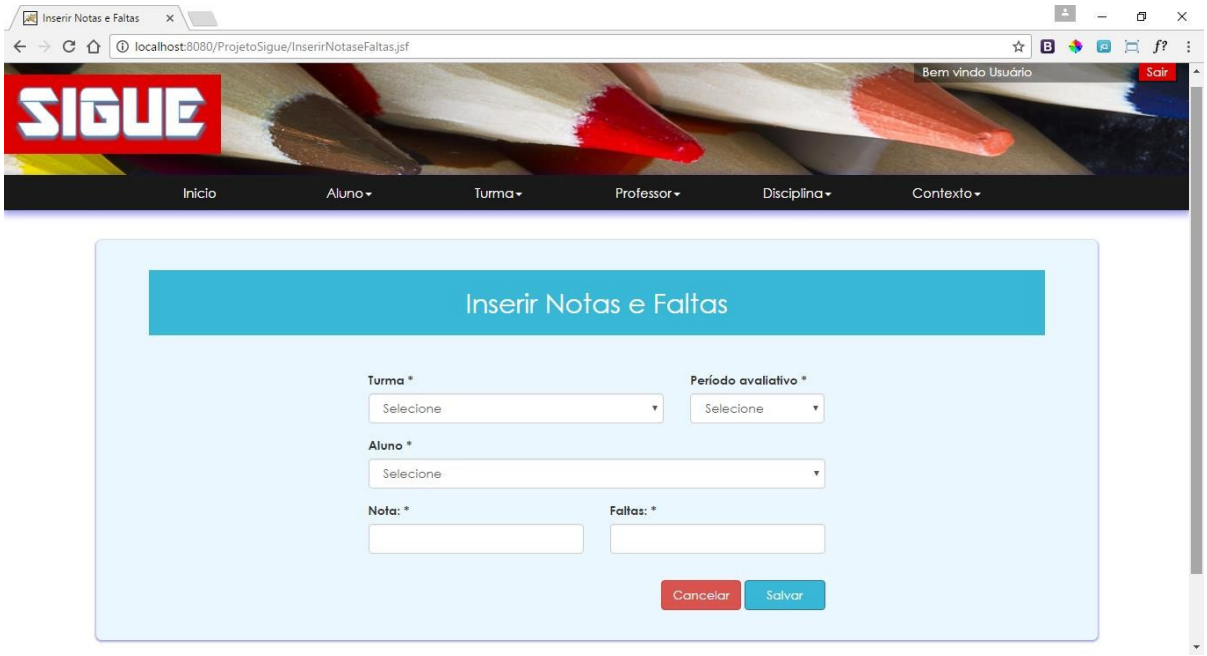
QUADRO 2 – Lista de requisitos não funcionais.

Cód.	Requisito
RNF-01	O acesso ao sistema será através de login, com usuário e senha pré-cadastrados.
RNF-02	Layout limpo
RNF-03	Fácil usabilidade
RNF-04	Os campos para preenchimento obrigatório deverão ser informados ao usuário
RNF-05	O sistema deverá ser web.
RNF-06	O sistema será desenvolvido com a linguagem Java.
RNF-07	Os dados devem estar protegidos de acesso não autorizado.
RNF-08	Os dados deverão ser armazenados em uma base íntegra.
RNF-09	Mostrar mensagem de confirmação ao realizar alguma operação que a requeira

Fonte: Elaborado pelo autor.

APÊNDICE E – Telas do sistema

FIGURA 13 – Tela de insersão de notas e faltas do aluno

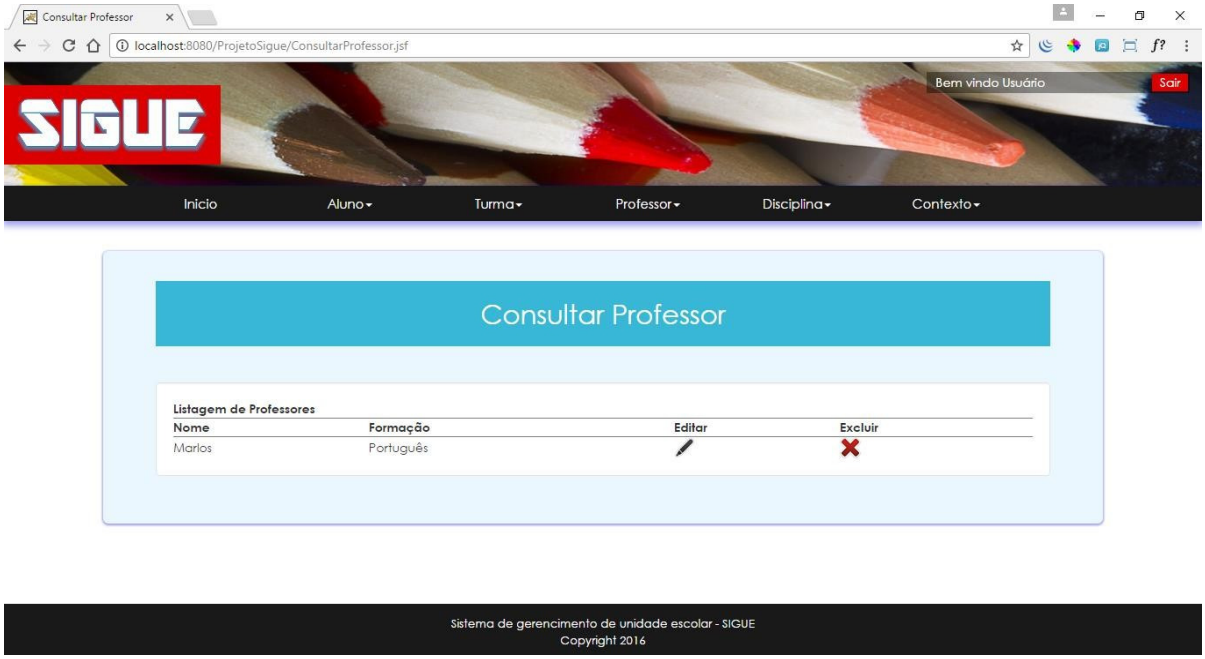


Fonte: Elaborada pelo autor.

FIGURA 14 – Tela de consulta de turma

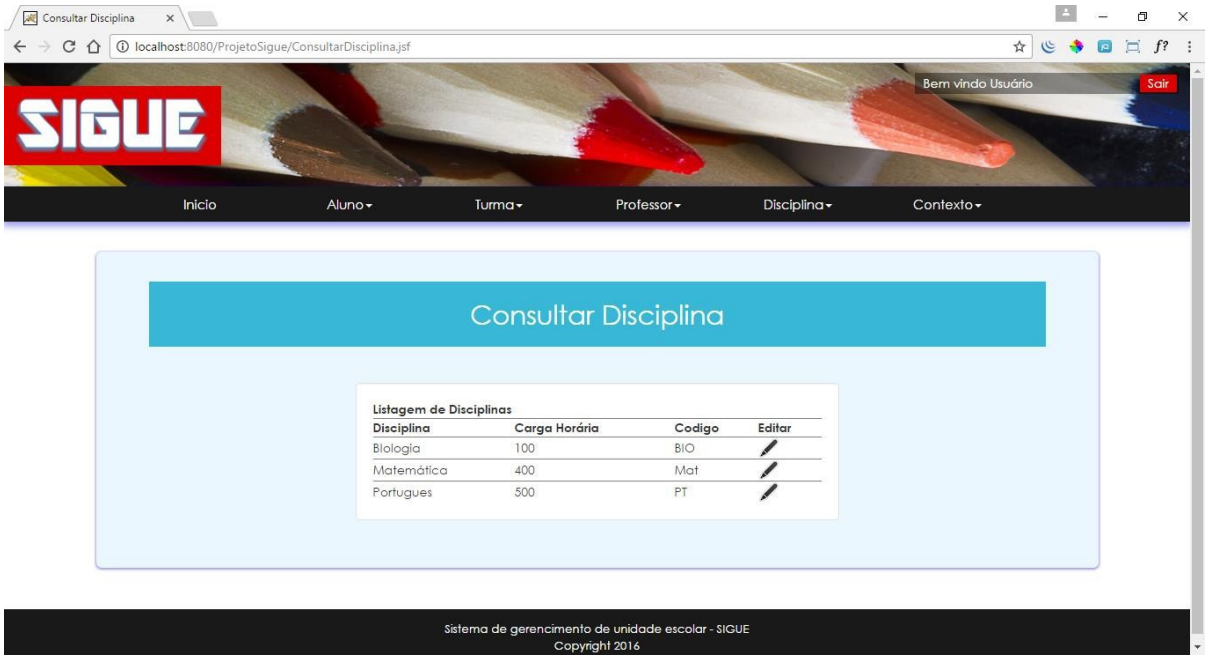
Fonte: Elaborada pelo autor.

FIGURA 15 – Tela de consulta de professor



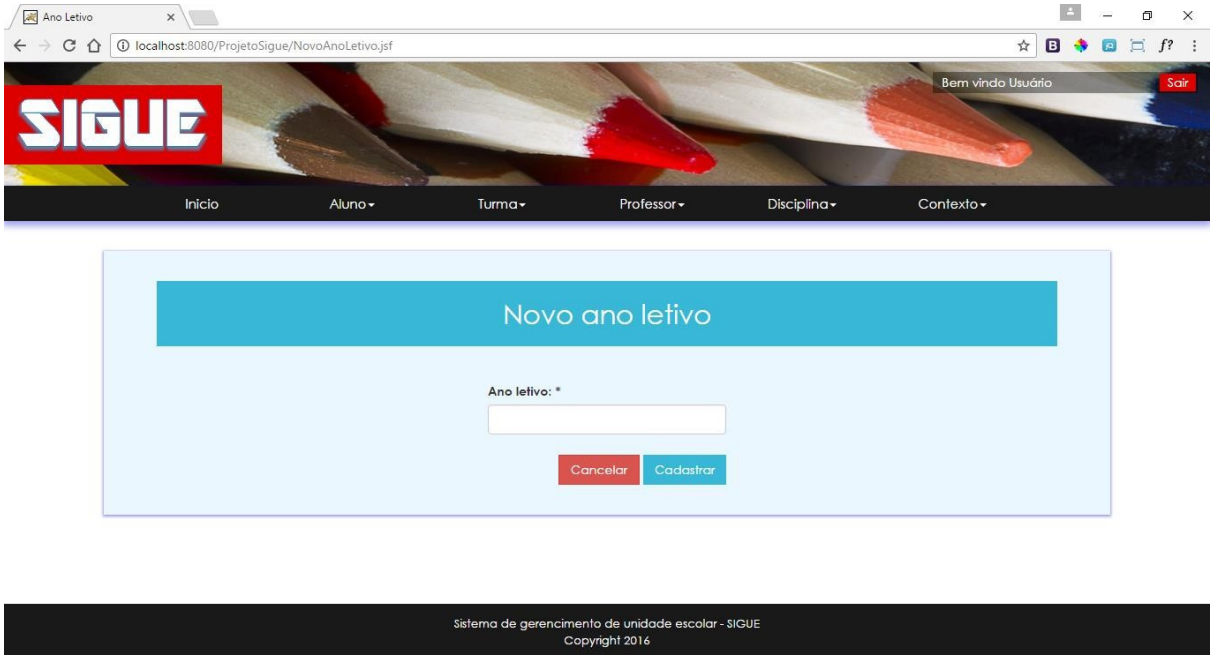
Fonte: Elaborada pelo autor.

FIGURA 16 – Tela de consulta de disciplina



Fonte: Elaborada pelo autor.

FIGURA 17 – Tela de cadastro de ano letivo

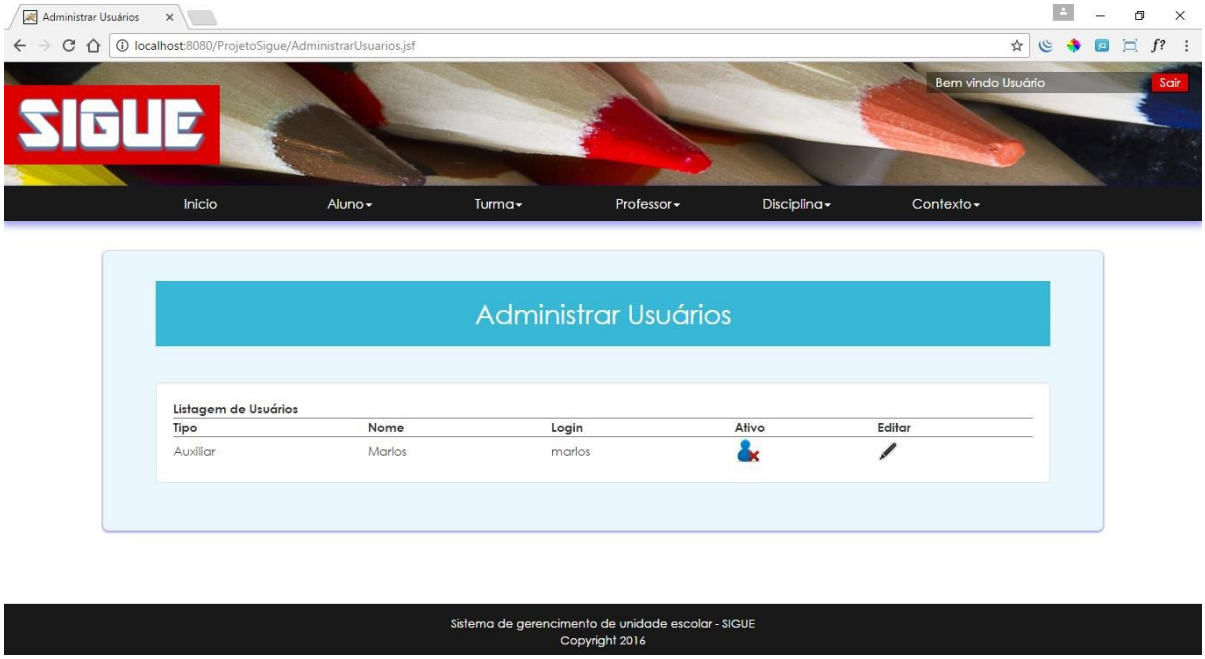


Fonte: Elaborada pelo autor.

FIGURA 18 – Tela de administração de ano letivo

Fonte: Elaborada pelo autor.

FIGURA 19 – Tela de administração de usuários



Fonte: Elaborada pelo autor.