

DESENVOLVIMENTO DO SISTEMA DE CONTROLE DE FALTAS (*WEB PRESENCE*)

Samuel Henrique Da Silva¹
Sérgio Augusto De Souza Moraes²

RESUMO

Com a globalização tomando conta cada vez mais do dia a dia das pessoas, tornando o uso da tecnologia um fator essencial para a comunicação e até mesmo a sobrevivência, é de extrema importância que todos estejam conectados, e por dentro das inovações tecnológicas no mundo. Pensando assim, este trabalho é o resultado de um estudo bibliográfico sobre as inovações tecnológicas voltadas para o ambiente da educação, que tem por objetivo auxiliar o professor à gerenciar o controle de faltas dos alunos de sua turma. Apresentando ao final deste trabalho a ferramenta web denominada "*Web Presence*", cujo demonstra o resultado de toda a pesquisa apresentado neste trabalho.

Palavras-chave: Tecnologia. Educação. Professor. *Web*.

ABSTRACT

*With globalization taking more and more account of people's daily lives, making the use of technology an essential factor for communication and even survival, it is of the utmost importance that everyone is connected, and within the technological innovations in the world. This work is the result of a bibliographical study on the technological innovations focused on the education environment, whose objective is to help the teacher to manage the control of the absences of the students of his class. Presenting at the end of this work the web tool called "*Web Presence*", which demonstrates the result of all the research presented in this work.*

Keywords: *Technology., Education. Teacher. Web.*

¹ Acadêmico do curso de Sistemas de Informação – Faculdade Atenas

² Docente do curso de Sistemas de Informação – Faculdade Atenas



INTRODUÇÃO

O uso da tecnologia é visível na atualidade, e vem crescendo cada dia a mais, tanto para pessoas comuns como empresas, uma vez que processos e atividades do cotidiano, como acessar transações bancárias via dispositivos móveis, compras online dentre outros estão fazendo parte do cotidiano das empresas. E como isso, percebe-se que esse crescente aumento vem possibilitando a ajudar a tomadas de decisões e também tornando os processos mais fáceis e ágeis (MOREIRA,2005).

E se tratando em tecnologia aplicada a educação pode-se dizer que é sinônimo de tecnologia educativa, pois a mesma se trata de uma função da mesma, seja qual for ela, e aos métodos envolvidos na atuação da educação, envolvendo também o método educativo ou instrutivo como propriamente dito. Tendo em vista esta realidade, e o ambiente em que se vive, alguns professores buscam a melhora e agilidade em suas atividades de ensino que optam pelo uso da tecnologia para auxilia-los (MIRANDA,2007).

A partir dessas informações o presente trabalho apresenta um sistema de controle de presença dos alunos de determinada escola, integrando a tecnologia às escolas, possibilitando mais agilidade em determinado processo.

TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO (TIC) NO AMBIENTE DE ENSINO

Com o crescente aumento das tecnologias de informação e comunicação na atualidade depara-se com a constante necessidade de otimizar e agilizar as tarefas do dia a dia. Nesse sentido, observa-se que o ambiente escolar também está em busca da utilização de tecnologias buscando otimizações e melhoramentos dos processos empresariais. E percebe-se a falta de alguns tipos de tecnologias no meio acadêmico proporciona um despreparo e até limitação na execução de atividades diárias, seja pelo motivo de várias instituições não investirem em sua infraestrutura ou por se ter tecnologias com alto custos (MORAN,2003).

Quando se usa TIC no ambiente de ensino, estima-se que esses recursos sejam capazes de proporcionar soluções didáticas mais rápidas,

melhorando assim a qualidade do ensino. Acredita-se que uma escola, precisa estar em um projeto de ação, para que assim tenha o trabalho de incentivo as mais diversas experiências, pois em certas situações pedagógicas permite a reelaboração e a reestruturação do processo de ensino-aprendizagem, e que são fortemente melhoradas com os recursos tecnológicos disponíveis (BRITO,2006).

Por outro lado, observa-se que na busca para introduzir as novas tecnologias no ambiente escolar, e em alguns casos, não se capacitam os professores, exigindo dos mesmos que aprendam a manipular os *softwares* e avaliar a nova maneira que está sendo utilizada para o ensino. Dessa forma, é preciso que a inclusão da tecnologia nas escolas tanto por parte dos professores quanto direção e alunos, busquem ideias e capacitações para melhorar o ensino (MERCADO,2002.).

Por isso é necessário que os envolvidos estejam cientes que após a inclusão de tecnologias serão desenvolvidos conceitos novos, para que seja possível aprender e aplicar uma nova forma a ensinar, de forma presencial ou não. E um aspecto que tem contribuído para tais evoluções e construção novos horizontes, tem-se o ensino através da internet (LAUDON,2005).

SISTEMAS DE INFORMAÇÃO

Observa-se que certas empresas nos últimos anos estão necessitadas de fazerem um *upgrade* em sua infraestrutura e processos, para poderem variar o seu jeito de trabalhar a fim de proporcionar mais agilidade as necessidades dos clientes, e assim, identifica-se que a organização tem contado com o apoio de sistemas de informação computadorizados (MORAN,2004). Exibe-se, segundo Nogueira (2004) que sistemas de informação hoje em dia estão se tornando importantes no mundo dos negócios, pois estão transformando as metodologias e processos das empresas, bem como a forma de pensar e trabalhar, pelo motivo de existir várias ferramentas em tecnologias (NOGUEIRA,2004).

E assim, é preciso compreender o conceito de sistemas de informação, que segundo O'Brien (2010), "é um conjunto organizado de pessoas, *hardware*, *software*, redes de comunicações e recursos de dados que coleta, transforma e dissemina informações em uma organização". Com isso nota-se a extrema relevância o sistema de informação para o dia a dia. E assim, observa-se que os

sistemas de informação podem estar associados ao uso de recursos de informática.

Nesse sentido a informática é uma grande ferramenta aliada no mundo atual, e é de extrema importância que as organizações e as pessoas, que ainda não se adaptaram com os parâmetros, por qualquer motivo sendo específico ou não, podem começar a reavaliar os conceitos de suas atividades e processos, pois certamente podem não ter controle por causa da falta desses recursos, e como consequência pode acarretar em prejuízos para a empresa. Percebe-se então que essas ferramentas são de grande importância para ajudar a tomadas de decisão desde pessoal à corporativa (BALTZAN,2012.).

As empresas que não fazem o uso de sistemas da informação computadorizados, podem correr um sério risco de serem deixadas para trás pelas organizações que fazem o uso de recursos tecnológicos. Por isso, as organizações devem procurar se familiarizar com o crescente avanço da tecnologia, para assim conseguir acompanhar o veloz ritmo que acontece em nosso ambiente (MUÑOZ,2012).

SOFTWARE NA EDUCAÇÃO

Um *software* educativo é toda aplicação computacional que foi projetada exclusivamente para a utilização no ambiente da educação. Observa-se que a definição é ampla e pode abranger diversas áreas do ambiente de ensino em atividades presenciais, semipresenciais e atividades online. Existem quatro fatores que diferenciam um *software* comum de um software da área da educação: a fundamentação pedagógica, conteúdo, interação aluno-*software*-educativo-professor e a codificação (BASSANI,2006.).

Vale apresentar que, quando se usa um computador, por meio de um software, como ferramenta educativa, a utilização de tais recursos, tem sido proveitoso na rotina de aprendizagem dos alunos. Deve-se frisar que um software educativo não deve ser tratado como algo independente da orientação de professores. Ao utilizar teorias construtivas, onde o aluno é convidado a proporcionar sua própria aprendizagem, formulando o seu conhecimento através de sua interação com o meio educativo. Nesse caso o software educativo não será o centro das atenções, ele seria um parceiro intelectual do aluno que estimularia o



pensamento crítico, levando assim a sua aprendizagem (EICHLER,2006).

Quando se desenvolve um software voltado para a educação, umas das etapas primordiais de sua produção é definir a sua concepção pedagógica. E para isso, ter um ou vários pedagogos na equipe do projeto desde o seu desenvolvimento a sua implementação seria indispensável, para se ter uma ideia para iniciar o projeto (VALENTE,1999).

Portanto, para um bom desenvolvimento de um *software* para a área da educação é necessário ter um bom levantamento de requisitos.

DESENVOLVIMENTO DE SOFTWARE

SOFTWARE

Software é um termo inglês que significa brando ou suave, que se refere à um equipamento ou suporte lógico que são necessários para realizar um trabalho especial. Estes componentes lógicos, incluem aplicações informáticas como, um processador de textos que lhe permite realizar tarefas como edição de textos. *Software* é todo sistema que pode ser um sistema operativo, e que a sua função é facilitar a interação entre homem e máquina (MARÔCO,2010.)

Um *software* normalmente é construído em diversas funções, bibliotecas e módulos, que combinados geram um executável, e que quando executado, recebe algum tipo de entrada (*input*), logo em seguida o *software* realiza o processamento das informações seguindo uma serie de algoritmos que nele foi programado e assim libera um tipo de saída (*output*), que é o resultado do processamento. Um *software* bem desenvolvido passa por várias etapas, como a análise de requisitos, especificação, codificação, testes, documentação, treinamento, manutenção e a sua implantação (DE SOUZA SACERDOTE,2016).

LINGUAGENS: PROGRAMAÇÃO, ESTILIZAÇÃO E MARCAÇÃO

PHP

A linguagem de programação *php*, que teve origem no outono de 1994 por Rasmus Lerdorf, que originalmente era formada por um conjunto de *scripts* escritos em linguagem *C*, voltado para a criação de páginas dinâmicas. Com o

tempo, mais pessoas passaram a utilizar estes recursos, e com isso Rasmus adicionou outras series de recursos como a interação com o banco de dados. E em 1995 o código fonte do *php* foi liberado, e com isso, vários outros desenvolvedores começaram a usar o *php*. E hoje em dia, é uma linguagem muito conhecida usada para a programação *web* (DALL'OGGIO,2015).

CASCADING STYLE SHEETS (CSS)

Toda a estilização de uma página *web*, pode ser feita com o *Cascading Style Sheets (css)*, que é uma linguagem de folhas de estilo, utilizada para definir apresentações de documentos escritos em uma linguagem de marcação, e o seu principal objetivo é a separação entre o formato e o conteúdo de um documento. Para facilitar, em vez de colocar a formatação dentro do documento, o desenvolvedor cria um *link* que faz a ligação entre o *css* e a linguagem de marcação, e os estilos que estiverem no *css* é interpretada e exibida pelo navegador (AMARAL,2006).

HYPertext MARKUP LANGUAGE (HTML)

E o pôr fim a linguagem de marcação, o *HyperText Markup Language (html)*, que é uma linguagem utilizada para desenvolvimento de páginas *web*, e é interpretada por navegadores, que é composta por *tags*, que em conjunto com o *css*, e uma linguagem de programação como o *php* pode-se desenvolver várias aplicações *web* (MARCONDES,2005).

ENGENHARIA DE SOFTWARE

Quando se pensa no processo de criação de um *software*, deve imaginar que esse processo é feito em várias etapas, e uma das mais importantes delas, é a engenharia de *software*, que se baseia em um conjunto de regras básicas que monitoram cada área da tecnologia e também inclui atividades como a modelagem e também outras técnicas descritivas (SOMERVILLE,2003).

Nesse sentido a engenharia de software vem para garantir também que

o *software* deverá executar exatamente o que foi planejado, programado e projetado. E qualquer outra situação que não esteja documentada poderá provocar falhas no sistema, gerando assim perdas tanto no nível econômico quanto no humano (GIRARDI,2004).

LEVANTAMENTO DE REQUISITOS

O desenvolvimento de *software* é um item que depende diretamente do levantamento de requisitos, ou seja, as informações básicas de processos e atividades que darão como produto final um sistema computacional. E assim, o levantamento de requisitos contempla um ápice de operações associadas a gerencia de requisitos. Nessas atividades, pode-se listar: analisar, documentar, validar requisitos, gerencia-los, dentre outras atividades (BASSANI,2006).

A fase de levantamento de requisitos, é o momento em que se deve analisar ao máximo os objetivos principais que poderá ser implementado no sistema, assim observando também os problemas descritos pelos clientes. No segundo momento, após feita a coleta de requisitos, é necessário identificar as possíveis soluções e também os objetivos do problema em questão. Durante essa fase, faz-se necessário manter o diálogo com o cliente, para entender o real problema e suas respectivas necessidades (BARBOSA,2009).

Os problemas que surgem com o levantamento de requisitos nem sempre são solucionados com a abordagem tecnológica, pois ressalta-se que o conjunto é formado também por aspectos sociais, informações e contextos que são de suma importância para um bom levantamento de requisitos (GUEDES,2007). Dessa maneira, entende-se que o levantamento dos requisitos bem-feita, influenciara diretamente na qualidade final do software.

REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

REQUISITOS FUNCIONAIS

Na engenharia de *software*, um requisito funcional é tudo aquilo que define uma função de um sistema e ou componente de um *software*. Uma função é retratada como um sistema de entradas, o seu comportamento e a sua saída.



Estes requisitos também podem ser cálculos que o sistema realiza, detalhes técnicos, manipulação de dados e processamentos, dentre outros. E todo requisito funcional é suportado por um requisito não funcional (ROMAN,1985.)

Requisito funcional se trata de uma ou mais funções que um *software* deverá realizar e ou atender. Ou seja, é uma solicitação, desejo ou necessidade que o *software* devesse materializar. É comum de vários profissionais da engenharia de *software* associarem a ideia de um requisito funcional a uma tela, uma rotina, que no fim serão as funcionalidades reais do sistema (SILVEIRA et al. ,2007).

REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais é todo requisito do sistema relacionado ao uso da aplicação, em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade e a manutenção de tecnologias envolvidas (CYSNEIROS, 1997).

ROSA (2005) diz que um requisito não funcional é uma não função do sistema, mas que precisa ser realizado para que o sistema consiga atender o seu propósito. Existe uma definição que diz que o requisito funcional diz o que o sistema irá fazer, e os requisitos não funcionais diz como o sistema irá fazer. Um RNF (requisito não funcional) tem o objetivo de atender os requisitos do sistema que não são funcionais, mas que fazem parte do escopo do projeto.

MODELAGEM DE SOFTWARE

Modelagem de *software* é toda representação de uma forma simplificada de algo da realidade. Pensando-se em uma analogia à uma planta de uma casa, ela de uma forma simplificada descreve toda a casa, desde os seus cômodos, banheiro, cozinha, sala. E quando se trata de um software também deve-se identificar o que o futuro sistema irá fazer (FILGUEIRA,2002.).

Lima (2006) diz que a modelagem de *software* é a atividade de construir modelos que expliquem as características e ou o comportamento de um sistema. E nessa construção do *software* os modelos podem ser usados para identificar as características e funcionalidades que o *software* irá ter. Uma forma muito comum da modelagem de *softwares* que não são orientados a objeto é através de fluxogramas, enquanto os *softwares* orientados a objeto normalmente utilizam a

UML (LINGUAGEM UNIFICADA DE MODELAGEM).

METODOLOGIA DE DESENVOLVIMENTO

A metodologia de desenvolvimento de *software* permite definir quem é o responsável por determinada tarefa, o que irá fazer, como irá fazer e define prazos para cada tarefa, das pessoas que estão envolvidas no projeto do *software*. (POMPILHO,2002).

Por várias vezes, o ato de usar uma metodologia de desenvolvimento de *software* é visto como a falta de criatividade dos responsáveis do projeto, como o acréscimo de burocracia, a exigência de vários documentos, e isso muitas das vezes é visto como desnecessário para se construir uma aplicação. A metodologia não deve limitar a criatividade profissional, ao contrário, ela deve auxiliar, ser um instrumento que ajude a coordenar as áreas e atividades envolvidas. (REZENDE,2002.).

Ao escolher a metodologia que será utilizada em determinado projeto, deverá ser seguida com base na natureza do projeto à ser desenvolvido, dos métodos e as ferramentas que serão utilizadas e do produto e até mesmo do público alvo (DOS SANTOS SOARES,2004).

LINGUAGEM UNIFICADA DE MODELAGEM (UML)

A UML se trata de uma unificação de varia notações gráficas, utilizada para modelar sistemas computacionais por meio do Paradigma de Orientação a Objetos. Essa linguagem se tornou, no ultimo anos, a linguagem-padrão de modelagem de software adotada internacionalmente pela indústria de Engenharia de Software pela sua hierarquia de diagramas que auxilia na estruturação do projeto, ajudando a organizar e agilizar o desenvolvimento de um determinado projeto (COSTA, 2006).

Para Bezerra (2006 p. 02), UML é uma linguagem gráfica para visualizar, especificar a construção e a documentação de um projeto, e ela proporciona uma forma padrão que vai desde a preparação ao desenvolvimento da documentação do projeto. A UML pode ser usada por qualquer membro da equipe

envolvida no projeto, desde a coleta de requisitos a implantação e manutenção do sistema, em todas as fases, mas principalmente no desenvolvimento do banco de dados.

BANCO DE DADOS

Muitas empresas lidam com um número grande de fluxo de dados, e pensando nisso, para ajuda-las a gerenciar esse grande fluxo de dados de uma maneira eficaz, surgiu o banco de dados que surgiu na década de 60, quando empresas ainda guardavam seus dados em papéis, e se depararam com a complexidade de localizar um dado procurando em pastas e arquivos físicos, e assim o banco de dados foi criado e permitiu guardar estes dados para que posteriormente possam ser consultados e analisados de uma maneira ágil (TURBAN,2002).

Para Date (2003 p.10) um banco de dados é uma biblioteca de dados que se relacionam de forma que crie um sentido, usada por uma aplicação de uma determinada empresa. Por outro lado, Para Turban (2002 p.591) banco de dados é um agrupamento lógico e organizado de arquivos inter-relacionados. Em um banco de dados, os dados são integrados e relacionados de tal forma que um conjunto de programas fornece acesso aos dados. Sendo assim, a criação de um banco de dados se divide em três etapas: modelagem conceitual, modelagem lógica e modelagem física. Na construção de um banco de dados, deve-se considerar características específicas da SGBD (Sistema de Gerenciamento de Banco de Dados), onde serão implementadas a base de dados.

SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Um sistema de gerenciamento de banco de dados (SGBD), basicamente, é um sistema de manutenção de registros, em outras palavras é um sistema computadorizado que permite que usuários armazenar, visualizar, atualizar, deletar, manipular as informações da melhor forma que bem entender. Essas informações pode ser qualquer coisa que tenha um significado ou algum valor ao indivíduo ou organização, ou seja, qualquer coisa que sirva para auxiliar o

processo geral das atividades do usuário (DATE,2004).

Segundo Costa (2006, p. 6) “usualmente, o projeto de banco de dados é baseado na construção de modelos e se divide em três etapas: modelagem conceitual, modelagem lógica e modelagem física. Que através dos vários modelos, busca descrever as informações que são armazenadas no banco de dados”. A modelagem de dados é uma técnica usada para a especificação das regras de negócios e as estruturas de dados de um banco de dados

MYSQL

De acordo com Suehring (2002, p. 25), o Programa de Banco de Dados *MySQL* é um sistema cliente/servidor que consiste de um servidor *SQL* multitarefa que suporta acessos diferentes, diversos programas, clientes e bibliotecas, ferramentas administrativas e diversas interfaces de programação (API's). O Servidor *MySQL* como uma biblioteca multitarefa que você pode ligar à sua aplicação para chegar a um produto mais rápido, menor e mais fácil de ser gerenciado.

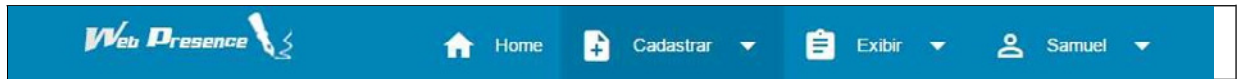
Segundo o site oficial do *MySQL*, ele possui um modo avançado de segurança, quando começa desde a sua inicialização, solicitando uma senha de usuário, essas senhas são todas criptografadas, o *mysql* criptografa as senhas através de um algoritmo parecido com o processo de autenticar o *login* do *Unix*, quando se instala o *mysql* ele contém dois usuários, um é o *root* que é usuário mestre e um usuário padrão. Para criar bancos de dados, tabelas e outros usuários você precisa estar logado como *root* (mestre) por isso ele é muito seguro e fácil de ser utilizado.

APRESENTAÇÃO DA FERRAMENTA DE AUXILIO NO GERENCIAMENTO DE FALTAS

A partir do levantamento bibliográfico, percebe-se que a utilização de software é um fator importante nas atividades diárias dos seres humanos. E tendo como referência essa premissa buscou-se o desenvolvimento de um sistema que fosse capaz de auxiliar no gerenciamento de faltas de uma escola. A partir dessas informações, desenvolveu-se o sistema *Web Presence*, cujo o foco busca trabalhar

os recursos, conforme a imagem:

FIGURA 01- MENU PRINCIPAL ADMINISTRADOR

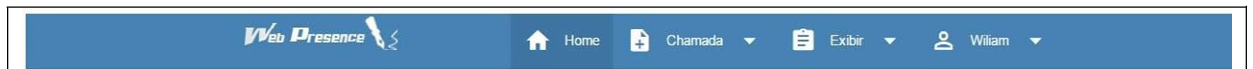


Fonte: Elaborado pelo autor.

A imagem acima se trata do menu principal do sistema do usuário administrador, onde é apresentado os botões, home, que volta para a página inicial, o botão cadastro, onde está localizado toda a parte de cadastro do sistema, como alunos, disciplinas, turmas, dentre outros, já no botão exibir é apresentado a lista de todos os cadastros realizados no sistema.

Já os usuários comuns, o menu do sistema é apresentado apenas os botões conforme a imagem abaixo, onde poderá apenas realizar a chamada e visualizar os dados do sistema.

FIGURA 02- MENU PRINCIPAL USUARIO

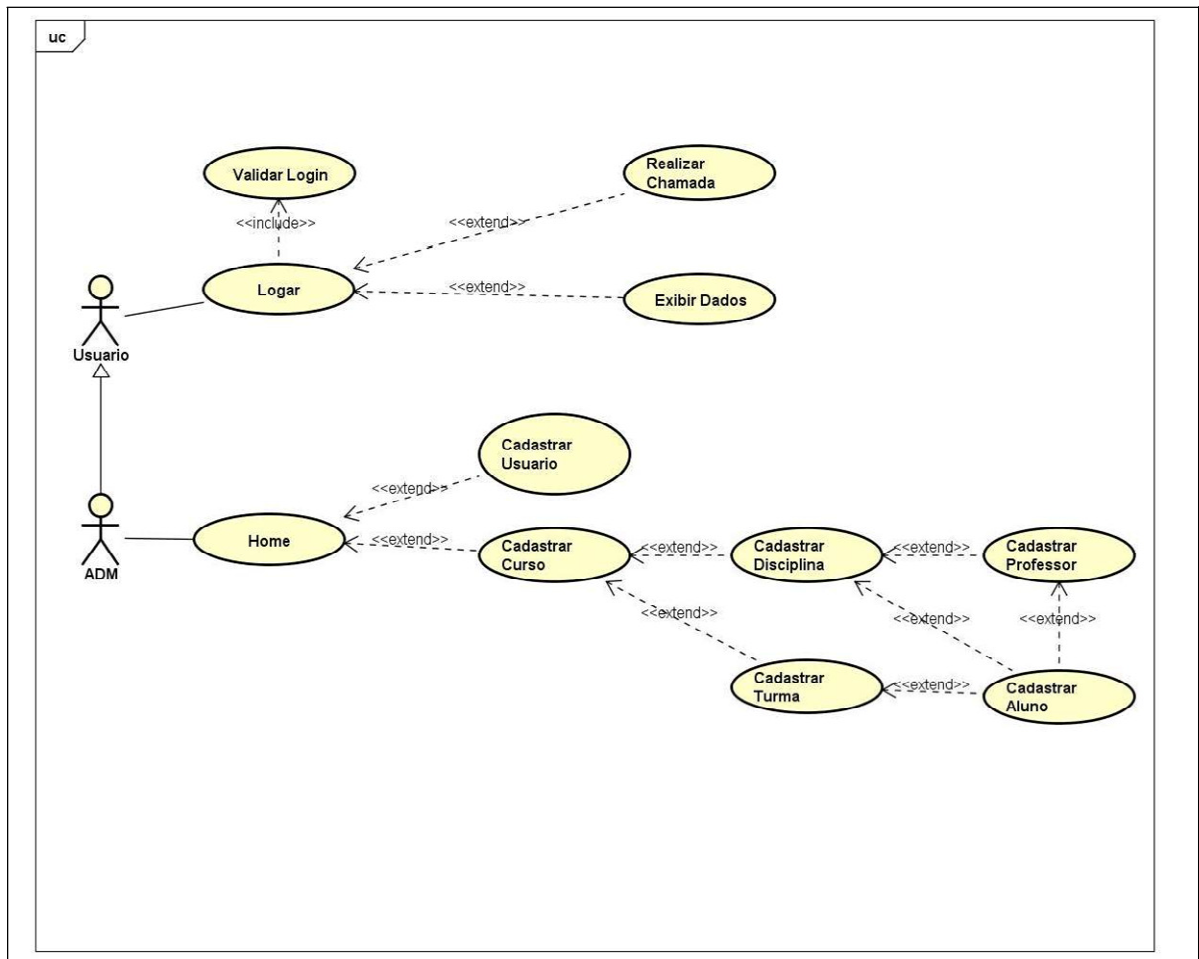


Fonte: Elaborado pelo autor.

Os recursos criados, surgiram a partir do levantamento de requisitos, e a por meio da captação dos requisitos, desenvolveu-se o diagrama de caso de uso, conforme mostra a figura 03. E na sequência, os requisitos e o diagrama possibilitaram a construção do sistema, que é descrito nas funcionalidades do diagrama de caso de uso a seguir, descrevendo de uma maneira narrativa como o sistema irá funcionar.

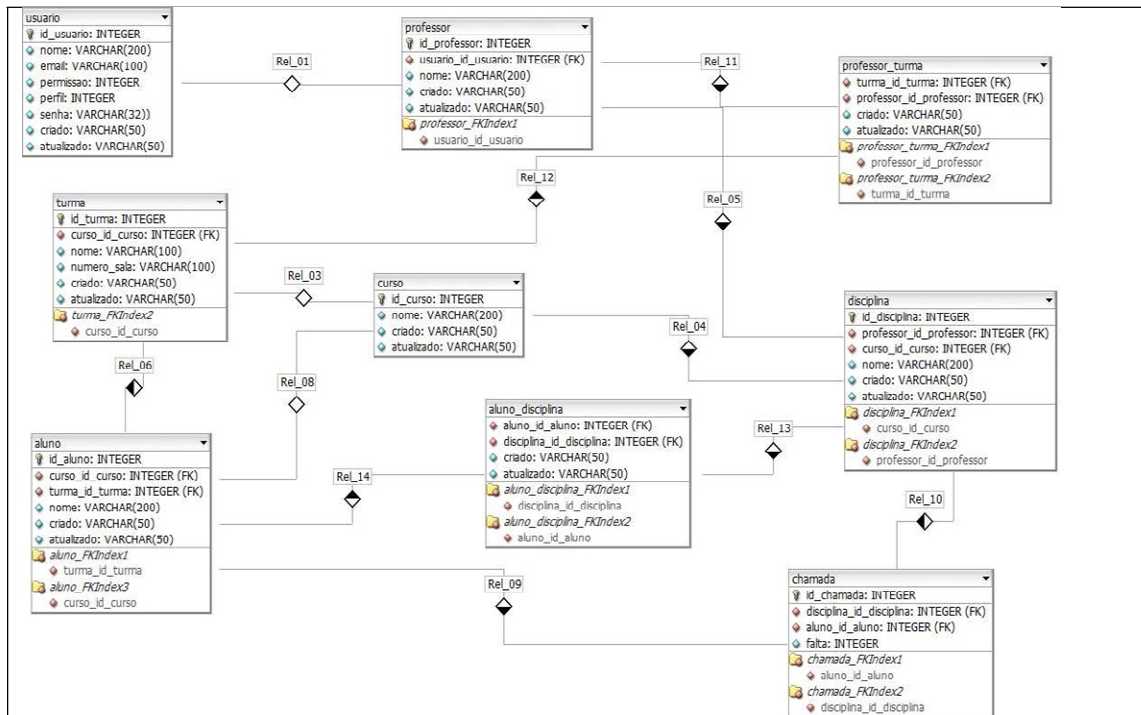
Contudo, cabe ainda apresentar que um sistema de informação precisa ser capaz de armazenar dados e fornecer informações. Nesse sentido e a partir dos requisitos foi elaborado o banco de dados, modelo lógico conforme a figura 03 a seguir:

FIGURA 03- DIAGRAMA DE CASO DE USO



Fonte: Elaborado pelo autor.

FIGURA 04- DIAGRAMA DE BANCO DE DADOS LÓGICO

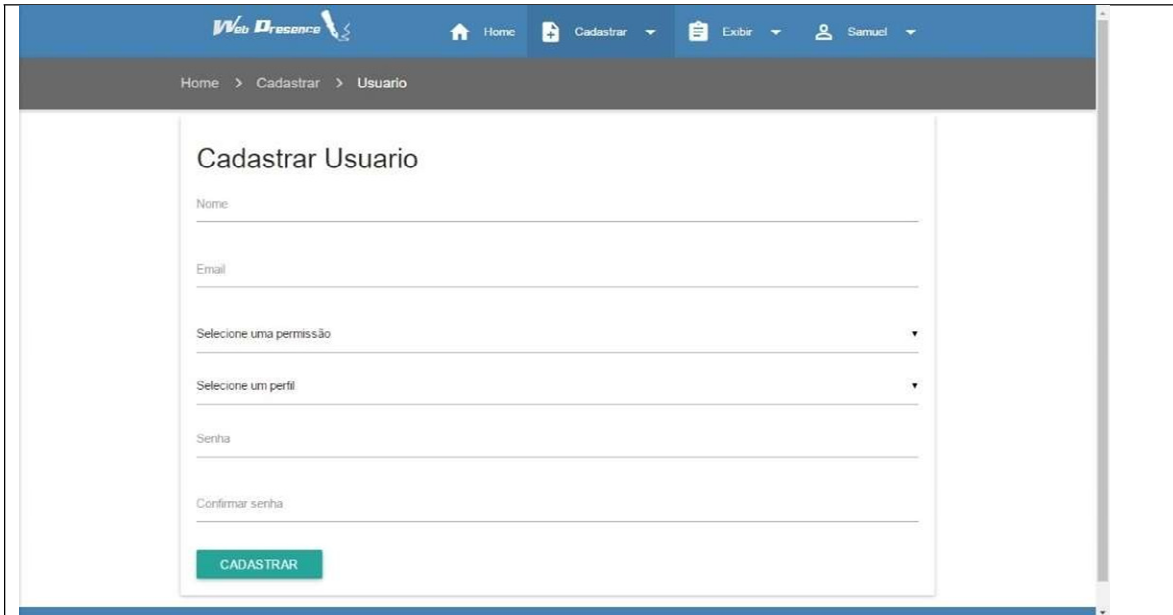


Fonte: Elaborado pelo autor.

Tendo elaborado o modelo lógico de banco de dados é desenvolvido na sequencia o *script* que permite a criação do banco propriamente dito, conforme mostra o apêndice A.

O processo principal do sistema é fazer o controle de presenças dos alunos de determinada escola, e para que o sistema, faça esse controle é necessário realizar os cadastros de professores, alunos, disciplinas e turma conforme figura 05 abaixo onde é exibido o cadastro de um usuário.

FIGURA 05- TELA CADASTRO DO USUARIO



Fonte: Elaborado pelo autor.

Quando se cadastra um usuário no sistema informa-se o nome, o e-mail, define a permissão do usuário, se ele será um administrador do sistema ou um usuário comum seleciona o perfil do usuário, se ele estará ativo ou inativo, e definimos uma senha. Ao se cadastrar um professor, informa-se o nome do professor, vinculamos ele a um usuário selecionando no campo, e seleciona-se as turmas que o professor irá lecionar.

E para acessar o sistema é necessário que o usuário administrador ou o usuário comum faça *login* conforme a figura 06 abaixo:

FIGURA 06 - TELA LOGIN

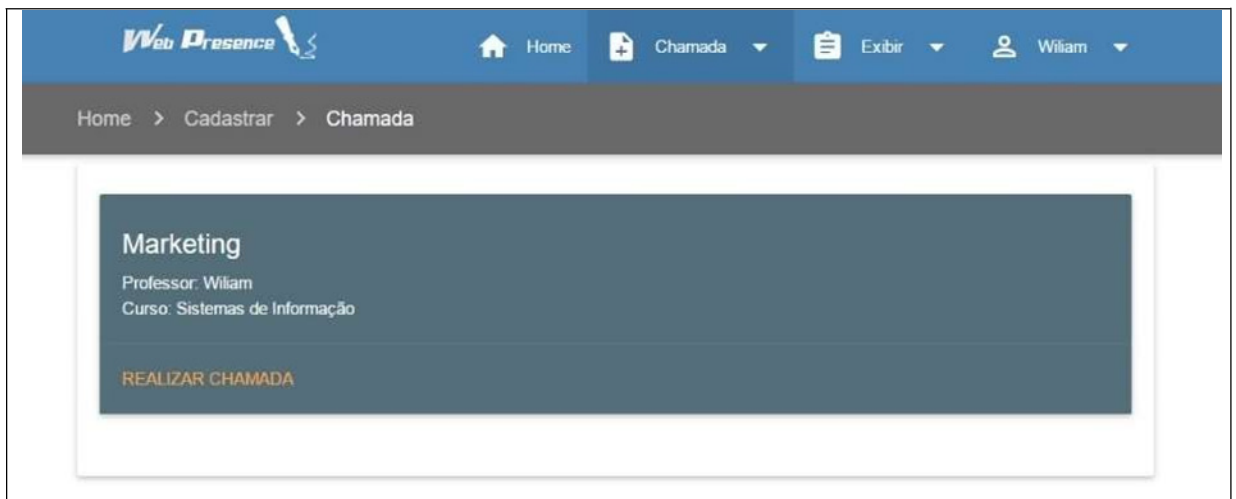


Fonte: Elaborado pelo autor.

Na sequência ao acessar o sistema, em sua tela principal o usuário terá acesso ao gerenciamento do curso, turma, número de salas, disciplinas e alunos. Sendo que no cadastro de curso usuário deverá informar apenas o nome do curso, já no cadastro da turma o usuário deverá informar o nome da turma e o número da sala de aula, após cadastrar a turma, o usuário deverá cadastrar a disciplina informando apenas o nome da mesma. Ao cadastrar um aluno o usuário colocará o nome, e selecionará qual curso e quais disciplinas o aluno irá ter.

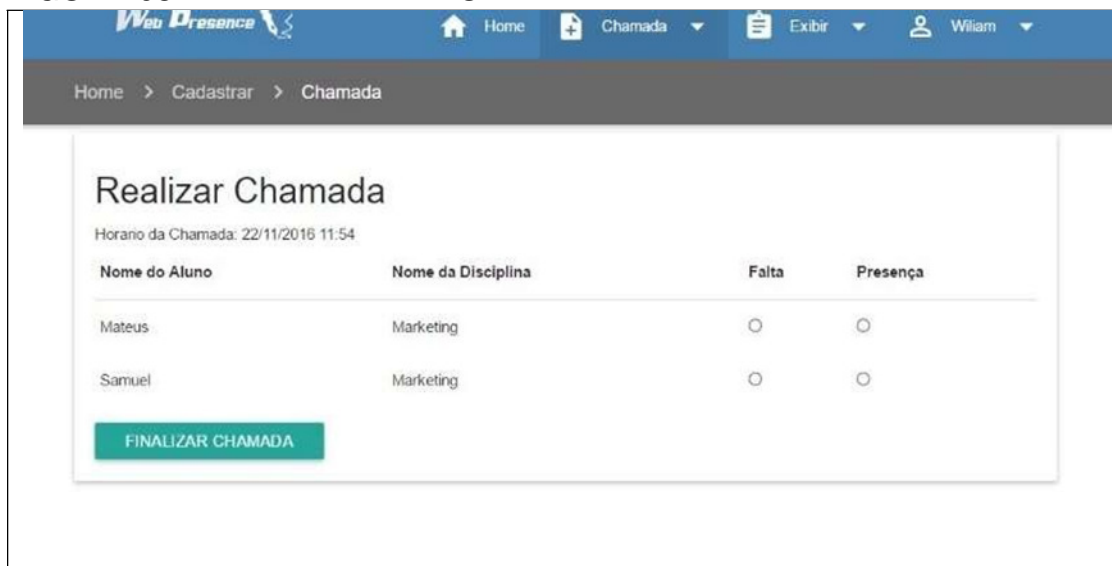
E por fim a realização da chamada conforme a imagem abaixo:

FIGURA 07 - TELA SELECIONAR DISCIPLINA



Fonte: Elaborado pelo autor.

Na figura 7, é exibida a tela a qual listará todas as disciplinas que foram vinculadas com o professor, após escolher a disciplina, basta selecionar a opção realizar chamada, e na sequência o usuário será direcionado para a tela com a lista dos alunos vinculados na disciplina para que possa fazer a chamada, conforme a figura 08:

FIGURA 08 - TELA REALIZAR CHAMADA

| Nome do Aluno | Nome da Disciplina | Falta | Presença |
|---------------|--------------------|-----------------------|-----------------------|
| Mateus | Marketing | <input type="radio"/> | <input type="radio"/> |
| Samuel | Marketing | <input type="radio"/> | <input type="radio"/> |

FINALIZAR CHAMADA

Fonte: Elaborado pelo autor.

CONSIDERAÇÕES FINAIS

Conclui-se que no mundo globalizado os avanços das tecnologias proporcionam a agilidade das atividades humanas, entretanto, há ainda a necessidade de desenvolvimento de recursos que sejam capazes de auxiliar o setor de educação, o qual se depara com um cenário um tanto quanto limitado. Com a ideia de levar a tecnologia no ambiente de ensino, entende-se que o presente trabalho propôs uma ferramenta que auxilie e gerencie a presença de alunos de uma instituição de ensino, de maneira ágil e pratica em sala de aula, alcançando-se assim os objetivos propostos.

Por fim, após todas as etapas de pesquisa apresentadas neste trabalho, estima-se o êxito da ferramenta, criando assim uma ferramenta *web* que traz funcionalidades para que o professor possa aproveitar mais o seu tempo em sala de aula e também possa usufruir, de uma tecnologia que permite que faça a chamada por qualquer dispositivo que acesse a internet, validando assim, a hipótese proposta no trabalho.

REFERÊNCIAS

AMARAL, Luis Gustavo. Guia de Consulta Rápida: CSS Cascading Style Sheets. **Sao Paulo: Novatec**, 2006.

BALTZAN, Paige; PHILLIPS, Amy. **Sistemas de informação**. AMGH Editora, 2012.

BARBOSA, Glívia et al. Um processo de elicitação de requisitos com foco na seleção da técnica de elicitação. **VIII Simpósio Brasileiro de Qualidade de Software**, p. 159-173, 2009.

BASSANI, Patricia Scherer et al. **Em busca de uma proposta metodológica para o desenvolvimento de software educativo colaborativo**. Renote, v. 4, n. 1, 2006.

BEZERRA, Eduardo. **Princípios De Análise E Projeto De Sistemas Com Uml-3ª Edição**. Elsevier Brasil, 2006.

BRITO, Glaucia da Silva. Tecnologias para transformar a educação. **Educar em Revista**, n. 28, p. 279-282, 2006.

CHAVES, Eduardo OC. Tecnologia na educação. **Encyclopaedia of Philosophy of Education, edited by Paulo Ghirardelli, Jr, and Michal A. Peteres. Published eletronically at**, 1999.

COSTA, Rogério Luis de Carvalho. **SQL Guia Prático**. 2 ed. Rio de Janeiro: Brasport, 2006.

COSTA, Rogério Luis de Carvalho. **SQL Guia Prático**. 2 ed. Rio de Janeiro: Brasport, 2006.

CYSNEIROS, Luiz Mareio; LEITE, J. C. S. P. Definindo requisitos não funcionais. **XI Simpósio Brasileiro de Engenharia de Software**. Fortaleza, CE, p. 33, 1997.

DALL'OGGIO, Pablo. **Php-programando com orientacao a objetos**. Novatec Editora, 2015.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. 8 ed. Rio de Janeiro: Elsevier, 2003.

DATE, Christopher J. **Introdução a sistemas de bancos de dados**. Elsevier Brasil, 2004.

DE SOUZA SACERDOTE, Helena Célia; FERNANDES, Jorge Henrique Cabral. Mediação da informação e mediação pedagógica: discussões conceituais. **Informação & Informação**, v. 21, n. 1, p. 407-425, 2016.

DOS SANTOS SOARES, Michel. 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.

EICHLER, Marcelo Leandro; DEL PINO, Jose Claudio. Carbópolis: um software para educação química. **Química: ensino médio. Brasília: Ministério da Educação. Secretaria de Educação Básica, 2006. p. 114-117, 2006.**

FILGUEIRA, João Maria; COSTA, Welbson Siqueira. **A Importância de utilizar UML para modelar sistemas**: Estudo de Caso. 2002.

GIRARDI, Rosario. Engenharia de Software baseada em Agentes. In:**Procedimentos do IV Congresso Brasileiro de Ciência da Computação (CBCComp 2004)**. 2004.

GUEDES, Gilleanes T. A. **UML 2: guia prático**. São Paulo: Novatec Editora, 2007.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação gerenciais: administrando a empresa digital**. Pearson Prentice Hall, 2005.

LIMA, Adailton et al. Gerência Flexível de Processos de Software com o Ambiente WebAPSEE. **XIII Sessão de Ferramentas–XX SBES**, 2006.

MARCONDES, Christian Alfim. HTML 4.0 Fundamental: a Base da Programação para WEB. **São Paulo: Érica**, 2005.

MARÔCO, João. **Análise de equações estruturais: Fundamentos teóricos, software & aplicações**. ReportNumber, Lda, 2010.

MERCADO, Luís Paulo Leopoldo. **Novas tecnologias na educação: reflexões sobre a prática**. UFAL, 2002.

MIRANDA, Guilhermina Lobato et al. Limites e possibilidades das TIC na educação.

Sísifo. Revista de Ciências da Educação, v. 3, p. 41-50, 2007.

MORAN, José Manuel. Gestão inovadora da escola com tecnologias. **Gestão educacional e tecnologia. São Paulo: Avercamp**, p. 151-164, 2003.

MORAN, José Manuel. Os novos espaços de atuação do professor com as tecnologias. **Conhecimento local e conhecimento universal: diversidade, mídias e tecnologias na educação. Curitiba: Champagnat**, v. 2, 2004.

MOREIRA, A. P.; MARQUES, L.; LOUREIRO, M^a. Percepções de professores e gestores de escolas relativas aos obstáculos à integração das TIC no ensino das ciências. **Enseñanza de las Ciencias**, n. Extra, p. 0001-5, 2005.

MUÑOZ, Paula Andrea Zuñiga. Sistemas de Informação. **TCC em Re-vista**, n. 15, 2012.

NOGUEIRA-MARTINS, Maria Cezira Fantini; BÓGUS, Cláudia Maria.

Considerações sobre a metodologia qualitativa como recurso para o estudo das ações de humanização em saúde. **Saúde e sociedade**, v. 13, n. 3, p. 44-57, 2004.

O'BRIEN, James A. **Sistemas de informação e as decisões gerenciais na era da Internet**. Tradutores: Célio Knipel Moreira, Cid Knipel Moreira; revisor técnico: Luiz Eduardo de Abreu Cunha; colaboração especial: Jakow Grajew. 3 ed. São Paulo: Saraiva, 2010.

POMPILHO, S. **Análise Essencial: guia prático de análise de sistemas**. Rio de Janeiro, Ciência Moderna, 2002.

RÉVILLION, Anya Sartori Piatnicki. A utilização de pesquisas exploratórias na área de marketing. **Revista Interdisciplinar de Marketing**, v. 2, n. 2, p. 21-37, 2015.

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 2. ed. Rio de Janeiro, Brasport, 2002.

ROMAN, G.-C. A taxonomy of current issues in requirements engineering. **Computer**, v. 18, n. 4, p. 14-23, 1985.

ROSA, M. C. S. Elicitação de Requisitos Funcionais e Não-Funcionais em Software Legado com Ênfase na Engenharia de Requisitos Orientada a Objetivos. **Monografia de graduação. Universidade Estadual do Oeste do Paraná. Cascavel-PR, Brasil, 2005.**

SILVEIRA, Naira Christofolletti et al. **Análise do impacto dos requisitos funcionais para registros bibliográficos (FRBR) nos pontos de acesso de responsabilidade pessoal**. 2007.

SOMERVILLE, Ian. **Engenharia de Software**. 6ª ed. São Paulo: Addison Wesley, 2003.

SUEHRING, Steve. **MySQL, A Bíblia**. 1 ed. Rio de Janeiro: Campus, 2002.

TURBAN, Efraim; RAINER JR., R. Kelly; POTTER, Richard E. **Introdução a Sistemas de Informação**. 4 ed. Rio de Janeiro: Editora Campus/Elsevier, 2002.

VALENTE, José Armando. Análise dos diferentes tipos de softwares usados na educação. **O computador na sociedade do conhecimento**, p. 71, 1999.

APÊNDICE A – SCRIPT DE CRIAÇÃO DE BANCO DE DADOS

E a partir do modelo lógico foi elaborado o modelo físico do banco de dados, abaixo:

```

CREATE TABLE aluno (
  id_aluno INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  curso_id_curso INTEGER UNSIGNED NOT NULL, turma_id_turma INTEGER UNSIGNED NOT NULL, nome
  VARCHAR(200) NULL,
  criado VARCHAR(50) NULL,
  atualizado VARCHAR(50) NULL, PRIMARY KEY(id_aluno),
  INDEX aluno_FKIndex1(turma_id_turma), INDEX aluno_FKIndex3(curso_id_curso)
);

CREATE TABLE aluno_disciplina ( aluno_id_aluno INTEGER UNSIGNED NOT NULL,
  disciplina_id_disciplina INTEGER UNSIGNED NOT NULL, criado VARCHAR(50) NULL,
  atualizado VARCHAR(50) NULL,
  INDEX aluno_disciplina_FKIndex1(disciplina_id_disciplina), INDEX
  aluno_disciplina_FKIndex2(aluno_id_aluno)
);

CREATE TABLE chamada (
  id_chamada INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  disciplina_id_disciplina INTEGER UNSIGNED NOT NULL, aluno_id_aluno INTEGER UNSIGNED NOT
  NULL,
  falta INTEGER UNSIGNED NULL,
  PRIMARY KEY(id_chamada),
  INDEX chamada_FKIndex1(aluno_id_aluno),
  INDEX chamada_FKIndex2(disciplina_id_disciplina)
);

CREATE TABLE curso (
  id_curso INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nome VARCHAR(200) NULL,
  criado VARCHAR(50) NULL,
  atualizado VARCHAR(50) NULL, PRIMARY KEY(id_curso)
);

CREATE TABLE disciplina (
  id_disciplina INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  professor_id_professor INTEGER UNSIGNED NOT NULL, curso_id_curso INTEGER UNSIGNED NOT
  NULL,
  nome VARCHAR(200) NULL, criado VARCHAR(50) NULL,
  atualizado VARCHAR(50) NULL, PRIMARY KEY(id_disciplina),
  INDEX disciplina_FKIndex1(curso_id_curso),
  INDEX disciplina_FKIndex2(professor_id_professor)
);

CREATE TABLE professor (
  id_professor INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  usuario_id_usuario INTEGER UNSIGNED NOT NULL, nome VARCHAR(200) NULL,
  criado VARCHAR(50) NULL,
  atualizado VARCHAR(50) NULL, PRIMARY KEY(id_professor),
  INDEX professor_FKIndex1(usuario_id_usuario)
);

CREATE TABLE professor_turma ( turma_id_turma INTEGER UNSIGNED NOT NULL,
  professor_id_professor INTEGER UNSIGNED NOT NULL,

```

```
criado VARCHAR(50) NULL,  
atualizado VARCHAR(50) NULL,  
INDEX professor_turma_FKIndex1(professor_id_professor), INDEX  
professor_turma_FKIndex2(turma_id_turma)  
);  
  
CREATE TABLE turma (  
id_turma INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
curso_id_curso INTEGER UNSIGNED NOT NULL, nome VARCHAR(100) NULL,  
numero_sala VARCHAR(100) NULL, criado VARCHAR(50) NULL,  
atualizado VARCHAR(50) NULL, PRIMARY KEY(id_turma),  
INDEX turma_FKIndex2(curso_id_curso)  
);  
  
CREATE TABLE usuario (  
id_usuario INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nome VARCHAR(200) NULL,  
email VARCHAR(100) NULL,  
permissao INTEGER UNSIGNED NULL, perfil INTEGER UNSIGNED NULL, senha VARCHAR(32) NULL,  
criado VARCHAR(50) NULL,  
atualizado VARCHAR(50) NULL, PRIMARY KEY(id_usuario)  
);
```

APÊNDICE B – LISTA DE REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS LISTA DE REQUISITOS FUNCIONAIS**QUADRO 1 – Requisitos Funcionais**

| Lista de Requisitos Funcionais | |
|---------------------------------------|---|
| RN1 | O sistema deverá cadastrar alunos e professores, capturando as informações através de um formulário. |
| RN2 | O sistema deverá calcular o percentual de faltas dos alunos de acordo com o limite de faltas definido pelo administrador; |
| RN3 | controlar acesso ao sistema, o usuário deverá inserir login e senha para ter acesso ao sistema; |
| RN4 | o sistema permitirá que o usuário realize consultas aos dados inseridos. |

Fonte: Elaborado pelo autor

QUADRO 2 – Requisitos não funcionais

| Lista de Requisitos Não Funcionais | |
|---|---|
| RNF1 | O sistema deverá apresentar dados confiáveis ao usuário; |
| RNF2 | portabilidade, o sistema será desenvolvido em uma plataforma <i>web</i> ; |
| RNF3 | segurança, para acessar o sistema o usuário deverá fazer <i>login</i> . |
| RNF4 | confiabilidade, o sistema deverá apresentar informações íntegras. |

Fonte: Elaborado pelo autor