



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Rodrigo Porfirio dos Santos

BLEAN
Ambiente Virtual de Aprendizagem

Americana, SP

2017



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Rodrigo Porfirio dos Santos

BLEAN
Ambiente Virtual de Aprendizagem

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Esp. Antonio Alfredo Lacerda.

Área de concentração: desenvolvimento de sistemas.

Americana, SP.

2017

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

S238b SANTOS, Rodrigo Porfirio dos

Blean : ambiente virtual de aprendizagem. / Rodrigo Porfirio dos Santos.
– Americana: 2017.

86f.

Monografia (Curso de Tecnologia em Análise e Desenvolvimento de
Sistemas) - - Faculdade de Tecnologia de Americana – Centro Estadual de
Educação Tecnológica Paula Souza

Orientador: Prof. Esp. Antonio Alfredo Lacerda

1. Educação à distância 2. Internet – rede de computadores I.
LACERDA, Antonio Alfredo II. Centro Estadual de Educação Tecnológica
Paula Souza – Faculdade de Tecnologia de Americana

CDU: 374
681.519

Rodrigo Porfirio dos Santos

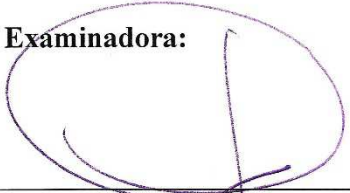
BLEAN
Ambiente Virtual de Aprendizagem

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.


Área de concentração: desenvolvimento de sistemas.

Americana, 29 de junho de 2017.

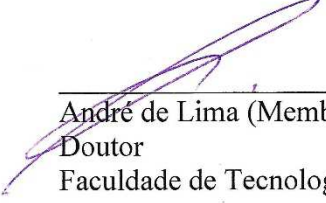
Banca Examinadora:



Antonio Alfredo Lacerda (Presidente)
Especialista
Faculdade de Tecnologia de Americana



Ana Lúcia Spigolon (Membro)
Especialista
Faculdade de Tecnologia de Americana



André de Lima (Membro)
Doutor
Faculdade de Tecnologia de Americana

AGRADECIMENTOS

Agradeço a todos os professores da Fatec Americana pelos conhecimentos e experiências compartilhadas, que ajudaram em minha formação profissional e amadurecimento pessoal. Agradeço especialmente aos professores que tiveram participação direta e indireta no desenvolvimento do Sistema Blean. Ao especialista Antonio Alfredo Lacerda por ensinar-me lógica de programação e compartilhar “minutos de sabedoria” com café e pão de queijo nos intervalos de aula. Ao mestre Diogenes de Oliveira, por ensinar-me a linguagem de programação C# e o ambiente de desenvolvimento web ASP.NET. Aos mestres José Alberto Florentino Rodrigues Filho e Wagner Siqueira Cavalcante por ensinar-me a desenvolver e gerenciar um banco de dados. Ao mestre Kleber de Oliveira Andrade, por ensinar-me a criar telas de programas somente por linhas de código. Ao mestre Luiz Rodolfo Barreto da Silva, por desatar o “nó na cabeça” causado por instruções de banco de dados e, juntamente com o mestre Thiago Salhab Alves, ensinar-me a engenharia envolvida no desenvolvimento de um sistema de informação. Ao mestre Benedito Luciano Antunes de França por ensinar-me a redigir as citações e referências presentes neste documento.

DEDICATÓRIA

Aos meus pais, por me mostrarem o valor do trabalho e por entenderem a importância da educação.

“O mérito é todo dos Santos. O erro e o pecado são meus. Mas onde está nossa vontade. Se tudo é vontade de Deus.”

Oswaldo Montenegro. Trecho da música: A lógica da criação.

RESUMO

O presente texto demonstra a criação de um Ambiente Virtual de Aprendizagem (AVA), batizado como Blean, utilizando-se principalmente dos recursos da linguagem de programação C# e do ambiente de desenvolvimento web ASP.NET. A escolha do desenvolvimento deste sistema deu-se porque um ambiente virtual de aprendizagem que, para sua utilização, não necessita da intervenção de um desenvolvedor e, tampouco da aquisição de um serviço de hospedagem web, estando acessível a qualquer pessoa que possua conexão com a internet, não é muito comum e também é difícil encontrar um destes ambientes em língua portuguesa. O nome do sistema, deu-se por conta do intuito de sua utilização, aprendizado misto

(*Blended Learning* ou *b-learning*), ou seja, utiliza-se o sistema para aulas presenciais com disponibilização de conteúdo on-line, incentivando a interação entre professor e aluno e também, incentivando o aluno a acessar as aulas fora do ambiente escolar, o que pode ser feito pela utilização de um *tablet*, *smartphone*, *notebook* ou computador pessoal. Neste projeto, realiza-se o estudo das tecnologias e metodologias utilizadas para o desenvolvimento do sistema, que são: educação a distância, internet, ambientes virtuais de aprendizagem, programação orientada a objeto, modelagem orientada a objeto, linguagem de programação C#, ambiente de desenvolvimento web ASP.NET, a Web e desenvolvimento de sites com a linguagem de marcação HTML e folha de estilo de exibição de conteúdo CSS e, o gerenciador de banco de dados MySQL. O projeto também apresenta ambientes virtuais de aprendizagem conhecidos por entidades de ensino superior, entidades de ensino técnico, professores e alunos, apresentando a forma de instalação ou contratação, utilização e principais recursos destes. O objetivo deste projeto foi alcançado com sucesso com o desenvolvimento e implementação do ambiente virtual de aprendizagem e o estudo das tecnologias e metodologias utilizadas.

Palavras Chave: Ambiente; Aprendizagem; Programação.

ABSTRACT

This text demonstrates the building of a Learning Management System (LMS), that is called Blean, mainly using the resources of C# programming language and ASP.NET web development environment. The choice for development of this system was because a learning management system, that for use, that don't want a web developer or purchase a web server, being accessible to anyone that has an internet connection, is not very common and it is also difficult to find one of these systems in Portuguese language. The name of the system was given by the intention of its use, blended learning or b-learning, that is the system is used for face-to-face classes with online content availability, encouraging interaction between teacher and student and also encouraging the student to access classes outside the school environment, which can be done by using a tablet, smartphone, notebook or personal computer. In this project, accomplish the technologies and methodologies used for the development of the system, which are: distance education, internet, learning management systems, object-oriented programming, object-oriented modeling, C# programming language, ASP.NET web development environment, the Web and sites development with the HTML markup language and CSS content display style sheet and, the MySQL database manager. The project also presents virtual learning environments known by higher education entities, technical education entities, teachers and students, presenting the form of installation or contracting, use and main resources of these. The objective of this project was successfully achieved with the development and implementation of the learning management system and the study of the technologies and methodologies used.

Keywords: *Environment; Learning; Programming.*

SUMÁRIO

LISTA DE FIGURAS	8
LISTA DE TABELAS	11
1 INTRODUÇÃO	12
2 REFERENCIAL TEÓRICO	15
2.1 EDUCAÇÃO A DISTÂNCIA	15
2.2 EVOLUÇÃO DA INTERNET	18
2.2.1 <i>A INTERNET NO BRASIL</i>	20
2.3 AMBIENTES VIRTUAIS DE APRENDIZAGEM	22
2.3.1 <i>MOODLE</i>	23
2.3.2 <i>TELEDUC</i>	27
2.3.3 <i>EDMODO</i>	28
2.4 PROGRAMAÇÃO ORIENTADA A OBJETOS	29
2.4.1 <i>JAVA</i>	40
2.4.2 <i>C#</i>	42
2.5 MODELAGEM ORIENTADA A OBJETOS	43
2.5.1 <i>DIAGRAMA DE CLASSES</i>	46
2.5.2 <i>DIAGRAMA DE OBJETOS</i>	47
2.5.3 <i>DIAGRAMA DE COMPONENTES</i>	48
2.5.4 <i>DIAGRAMA DE ESTRUTURA COMPOSTA</i>	49
2.5.5 <i>DIAGRAMA DE CASOS DE USO</i>	50
2.5.6 <i>DIAGRAMA DE SEQUÊNCIAS</i>	50
2.5.7 <i>DIAGRAMA DE COMUNICAÇÃO</i>	51
2.5.8 <i>DIAGRAMA DE GRÁFICO DE ESTADOS</i>	52
2.5.9 <i>DIAGRAMA DE ATIVIDADES</i>	53
2.5.10 <i>DIAGRAMA DE IMPLANTAÇÃO</i>	54
2.5.11 <i>DIAGRAMA DE PACOTES</i>	54
2.6 WEB	56
2.6.1 <i>Site</i>	57
2.7 MYSQL	59
2.8 .NET	60
2.8.1 <i>ASP.NET</i>	62
3 SISTEMA BLEAN	64
3.1 ESPECIFICAÇÃO DOS REQUISITOS	64

3.1.1	REQUISITOS FUNCIONAIS	64
3.1.2	REQUISITOS NÃO FUNCIONAIS	66
3.2	MODELAGEM	67
3.2.1	DIAGRAMA DE NAVEGAÇÃO.....	67
3.2.2	DIAGRAMA DE CASOS DE USO.....	68
3.2.3	DIAGRAMA DE CLASSES.....	69
3.2.4	DIAGRAMA DE OBJETOS	70
3.2.5	DIAGRAMA DE SEQUÊNCIAS.....	70
3.2.6	DIAGRAMA DE ATIVIDADES.....	72
3.2.7	DIAGRAMA DE PACOTES.....	73
3.2.8	DIAGRAMA DE IMPLANTAÇÃO	74
3.2.9	DIAGRAMA ENTIDADE RELACIONAMENTO (DER)	75
3.2.10	MODELO ENTIDADE RELACIONAMENTO (MER)	76
3.2.11	DICIONÁRIO DE DADOS	77
3.3	IMPLEMENTAÇÃO	79
3.4	IMAGENS DAS TELAS	80
3.5	RESULTADO.....	89
4	CONSIDERAÇÕES FINAIS.....	90
	APÊNDICE A – INSTRUÇÃO SQL (CÓDIGO FONTE).....	91
	APÊNDICE B – EXIBIÇÃO DE AULAS (CÓDIGO FONTE)	92
	REFERÊNCIAS	93

LISTA DE FIGURAS

Figura 1: Estrutura do sistema de telefonia.....	18
Figura 2: Sistema distribuído de comutação.....	19
Figura 3: Primeiro curso do Moodle	25
Figura 4: Exemplo de diagrama de classes	47
Figura 5: Exemplo de diagrama de objetos	48
Figura 6: Exemplo de diagrama de componentes com interfaces.....	49
Figura 7: Exemplo de diagrama de estrutura composta com conectores	49
Figura 8: Exemplo de diagrama de casos de uso	50
Figura 9: Exemplo de diagrama de sequências	51
Figura 10: Exemplo de diagrama de comunicação	52
Figura 11: Exemplo de diagrama de gráfico de estados	52
Figura 12: Exemplo de diagrama de atividades	53
Figura 13: Exemplo de diagrama de implantação.....	54
Figura 14: Exemplo de diagrama de pacotes	55
Figura 15: Diagrama de navegação	67
Figura 16: Diagrama de casos de uso	68
Figura 17: Diagrama de classes.....	69
Figura 18: Diagrama de objetos.....	70
Figura 19: Diagrama de sequências.....	71
Figura 20: Diagrama de atividades.....	72
Figura 21: Diagrama de pacotes	73
Figura 22: Diagrama de implantação	75
Figura 23: DER (diagrama entidade relacionamento).....	76
Figura 24: MER (modelo entidade relacionamento).....	76
Figura 25: Perfil visitante - tela inicial	81
Figura 26: Perfil visitante - login.....	81
Figura 27: Perfil visitante - cadastro de novo usuário	82
Figura 28: Usuário registrado - dados pessoais	82
Figura 29: Professor - criar ou editar cursos	83
Figura 30: Professor - criar ou editar turmas.....	85
Figura 31: Professor – criar aula ou editar o título da aula	85
Figura 32: Professor - criar ou editar o conteúdo da aula.....	86

Figura 33: Professor - gerenciar contas de alunos	86
Figura 34: Aluno - inscrever-se e visualizar turmas e aulas.....	87
Figura 35: Aluno - visualizar conteúdo da aula.....	88
Figura 36: Administrador: visualizar ou editar contas de usuários.....	88

LISTA DE TABELAS

Tabela 1: Tecnologias ASP.NET web dinâmicas.....	62
Tabela 2: Dicionário de dados - tabela "Curso"	77
Tabela 3: Dicionário de dados - tabela "Turma"	77
Tabela 4: Dicionário de dados - tabela "Turma_tem_aula"	77
Tabela 5: Dicionário de dados - tabela "Turma_tem_curso"	79
Tabela 6: Dicionário de dados - tabela "Usuario"	79
Tabela 7: Dicionário de dados - tabela "Turma_tem_usuario"	79

1 INTRODUÇÃO

A utilização de dispositivos eletrônicos (computadores, notebooks, smartphones e tablets) como “sala de aula” tem aumentado nos últimos anos com a popularização de preços dos dispositivos, aumento da capacidade de processamento dos mesmos e acesso à internet cada vez mais barato e com velocidades maiores. O método de ensino em que se pode estudar a qualquer hora e lugar, é conhecido como educação a distância (EAD) e, teve seu início com a utilização de envio e recebimento de material pelos sistemas de correspondência. Este mesmo método, quando se utiliza a internet, recebe o nome de *e-learning* (*eletronic learning*), que significa ensino eletrônico e, o conteúdo das aulas, pode ser disponibilizado através de um ambiente virtual de aprendizagem (AVA ou LMS – *Learning Management System*).

Algumas instituições de ensino implantam o ambiente virtual de aprendizagem junto ao site da instituição. A utilização do sistema ocorre de forma mista, dentro e fora da sala de aula, com o propósito de auxiliar o professor a disponibilizar conteúdo e tarefas para seus alunos. Então surge um novo termo: *b-learning*.

B-learning (*blended learning*) significa ensino misto, onde parte do conteúdo é disponibilizado on-line e outra parte em encontros presenciais. Como ambiente virtual de aprendizagem pode-se citar o Moodle, Edmodo e TelEduc. Estes ambientes são classificados como *e-learning* e têm ampla utilização em sala de aula por instituições de cursos superiores e cursos técnicos, portanto, mesmo classificados como *e-learning* podem ser utilizadas como *b-learning*. Estes ambientes são gratuitos e o acesso e utilização é feita, para o Edmodo, como uma rede social, para o Moodle, basta fazer o download no site do desenvolvedor e instalar em seu serviço de hospedagem web e, o TelEduc, além do download, permite a utilização através do servidor NIED (Núcleo de Informática Aplicada à Educação) da Unicamp, mas necessita de cadastro e aprovação, e seu uso é restrito a cursos sem fins lucrativos.

Para implantar um destes ambientes (Moodle ou Teleduc), é necessário que a instituição de ensino (ou professor) possua uma conta de hospedagem em um servidor web Apache e acesso a um sistema de gerenciamento de banco de dados (SGBD) de linguagem de consulta estruturada (SQL - *Structured Query Language*). O profissional responsável pela implantação deve saber utilizar o protocolo de transferência de arquivos (FTP - *File Transfer Protocol*), para transferir os arquivos de seu computador para o servidor web e a linguagem de script PHP (*Hypertext Preprocessor*), para integrar o sistema com o banco de dados e fazer alterações no ambiente. Instituições de ensino, em sua maioria, possuem um profissional com estas qualificações e também o serviço de hospedagem web, então, somente o professor que

trabalhe em uma instituição com estas características ou, ele próprio possua estes requisitos, pode utilizar das facilidades oferecidas por estas ferramentas. Já com o Edmodo, por não necessitar de instalação, o uso é mais simples, mas ele não está disponível na língua portuguesa.

O sistema Blean surge para oferecer aos professores, que não possuem os requisitos de implantação de um ambiente virtual de aprendizagem em suas aulas, a possibilidade de uso do sistema mesmo sem possuir um serviço de hospedagem web e sem necessitar de conhecimento em desenvolvimento web e banco de dados e, está disponível na língua portuguesa.

Este documento está dividido em quatro capítulos, o primeiro é uma introdução com a apresentação da metodologia de ensino a distância, ambientes virtuais de aprendizagem, objetivo e justificativa. O segundo apresenta o referencial teórico para o desenvolvimento do sistema, dividido em oito seções. A primeira seção apresenta a história da educação a distância, que surge com cursos disponíveis em jornais e entregues pelo sistema postal, depois rádio, televisão e chegando a internet. A segunda seção apresenta a evolução da internet, desde seu surgimento com o desenvolvimento do comutador de pacotes criado por Paul Baran em 1960 e utilizado pelo Departamento de Defesa dos Estados Unidos nos anos seguintes dando origem a ARPANET, até a chegada da Web em 1989, desenvolvida por Tim Berners-Lee e, a chegada da Bitnet ao Brasil no ano de 1988. A terceira seção apresenta três ambientes virtuais de aprendizagem gratuitos, utilizados por instituições de ensino e professores. A quarta seção apresenta as características da programação orientada a objetos, exibe alguns exemplos de implementação em linguagem Java e C#, apresenta também a história do desenvolvimento, particularidades de compilação e ambientes de desenvolvimento das duas linguagens. A quinta seção apresenta a linguagem de modelagem de sistemas UML, sua história, definições de modelo e exemplos de diagramas utilizados para a modelagem. A sexta seção apresenta a rede mundial de computadores, seu criador e sua motivação, a diferença entre Web e internet, os protocolos e a forma de escrita de um site através da marcação de texto e folhas de estilo. A sétima seção explica o que é um banco de dados e apresenta o sistema de gerenciamento MySQL. A oitava seção apresenta o .NET, um ambiente desenvolvido pela Microsoft para o desenvolvimento e execução de aplicações, seu método de compilação, o ambiente de desenvolvimento ASP.NET, e o programa Visual Studio. O terceiro capítulo apresenta o desenvolvimento do ambiente virtual de aprendizagem Blean, suas especificações, modelagem, implementação e imagens das telas do sistema. O quarto

capítulo apresenta as considerações finais sobre o trabalho de pesquisa e desenvolvimento do sistema.

O objetivo deste trabalho é o desenvolvimento de um ambiente virtual de aprendizagem e apresentação das tecnologias aplicadas no desenvolvimento deste. Para que o objetivo seja alcançado, faz-se necessário um estudo histórico do ensino a distância e evolução da internet, estudo de alguns ambientes virtuais de aprendizagem conhecidos e, estudo das tecnologias utilizadas para o desenvolvimento do novo sistema.

O sistema desenvolvido neste projeto, batizado como Blean, tem como justificativa oferecer as facilidades do ambiente virtual de aprendizagem aos professores que não possuem um sistema de hospedagem web e não possuem as habilidades necessárias para a implantação do mesmo. Levando-se em conta que a maioria desses ambientes são disponibilizados aos professores pela instituição de ensino, o sistema Blean exclui esta dependência, oferecendo ao professor a utilização do sistema pelo tempo que achar necessário sem necessitar da aprovação de sua instituição de ensino.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta um estudo das metodologias e tecnologias utilizadas no desenvolvimento do ambiente virtual de aprendizagem Blean. Este estudo apresenta uma relação de interação entre áreas de conhecimento distintas, buscando apresentar a importância de cada uma para o desenvolvimento do sistema desenvolvido.

2.1 EDUCAÇÃO A DISTÂNCIA

Esta seção apresenta a história da educação a distância, com os primeiros relatos nos Estados Unidos da América com a utilização do sistema postal, seguindo para o Brasil, com a utilização do sistema postal, rádio, televisão e internet.

De acordo com artigo 205 da Constituição da República Federativa do Brasil (1988), “a educação, direito de todos e dever do Estado e da família, será promovida e incentivada com a colaboração da sociedade, visando ao pleno desenvolvimento da pessoa, seu preparo para o exercício da cidadania e sua qualificação para o trabalho”.

Educação a distância tem sua origem na metade do século 19 na Europa e nos Estados Unidos. Os pioneiros da educação a distância usavam a melhor tecnologia daquele tempo, o sistema postal, para levar oportunidades de estudos para pessoas que queriam aprender, mas não estavam aptos a frequentar escolas convencionais. As pessoas mais beneficiadas pela educação por correspondência eram aquelas com deficiências físicas, mulheres que não tinham sua matrícula aceita em instituições educacionais abertas somente a homens, pessoas que trabalhavam durante o horário normal de aula, e aqueles que moravam em regiões remotas que não tinham escolas¹ (HISTORY D., 2011).

De acordo com Como (2016), a educação a distância tem seu primeiro registro em 1728 na cidade de Boston nos Estados Unidos, quando o professor Caleb Phillips anunciou no jornal local seu curso de taquigrafia², onde era oferecido material semanalmente por correspondência. Também é datado que em 1840, na Inglaterra, o professor Isaac Pitman iniciou seu curso de taquigrafia de passagens bíblicas por correspondência.

¹ Distance education traces its origins to mid-19th century Europe and the United States. The pioneers of distance education used the best technology of their day, the postal system, to open educational opportunities to people who wanted to learn but were not able to attend conventional schools. People who most benefited from such correspondence education included those with physical disabilities, women who were not allowed to enroll in educational institutions open only to men, people who had jobs during normal school hours, and those who lived in remote regions where schools did not exist. (HISTORY OF DISTANCE LEARNING, 2011. *Tradução nossa*)

² Técnica e método de escrever as palavras na mesma velocidade que são pronunciadas, por meio de sinais e abreviaturas. (Michaelis - Dicionário da Língua Portuguesa)

Conforme Alves (2007), no Brasil, tem-se noticiado que a educação a distância teve seu início antes de 1900, quando eram anunciados cursos de datilografia por correspondência, em jornais no Rio de Janeiro. A educação a distância com registro datado, ocorreu em 1923 pela iniciativa privada com a criação da Rádio Sociedade do Rio de Janeiro, que transmitia educação popular. Alguns anos depois, surge o Ministério da Educação e o Serviço de Radiodifusão Educativa.

Com este novo serviço, abriu-se a possibilidade e o interesse da iniciativa privada em criar rádios educativas, como exemplo pode-se citar a Universidade do Ar criada pelo Serviço Nacional de Aprendizagem Comercial (SENAC), o projeto Movimento Brasileiro de Alfabetização (MOBRAL) do Governo Federal, o Movimento Educação de Base da igreja Católica e A Voz da Profecia da igreja Adventista.

“A revolução deflagrada em 1969 abortou grandes iniciativas, e o sistema de censura praticamente liquidou a rádio educativa brasileira” (ALVES, 2007).

A censura estabelecida pelo regime militar ocorrido entre 1964 e 1984 fez com que as rádios que transmitiam conteúdo educacional parassem com suas transmissões, porém em 1972, junto com a televisão a cores, surge o Programa Nacional de Teleeducação (PRONTEL), vinculado a Secretaria Geral do Ministério da Educação e Cultura, que passou a regulamentar todo o material educativo distribuído por rádio, televisão e outros meios. Com a regulamentação, emissoras de televisão passaram a ter a obrigação de disponibilizar conteúdo educativo até o início da década de 1990.

Como exemplo, pode-se citar alguns programas e emissoras e seus respectivos mantenedores:

- TV Educativa – Ministério da Cultura;
- TV Escola – Ministério da Educação;
- Telecurso 2º Grau – Fundação Roberto Marinho;
- Canal Futura – Votorantim, FIRJAM, FIESP, CNI, Rede Globo, Fecomércio, Fundação Bradesco e Itaú Social e;
- Canal Universitário (CNU) – Universidade São Judas Tadeu, Pontifícia Universidade Católica de São Paulo (PUC), Universidade Presbiteriana Mackenzie, Universidade Paulista (UNIP), Instituto Nacional de Telecomunicações.

Vale destacar que, no Brasil, existe o Decreto nº 5.622, de 19 de dezembro de 2005, que regulamenta o artigo 80 da Lei nº 9.394 de 20 de dezembro de 1996, que estabelece no Artigo 1º:

Para os fins deste Decreto, caracteriza-se a educação a distância como modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos.

Na citação, pode-se observar o termo “tecnologias de informação” e a palavra “comunicação”, ambos precedidos da palavra “meios”. Isso ressalta que a educação a distância pode ser disponibilizada por meios de comunicação como, correspondência, rádio e televisão, e por meio de tecnologias de informação como, disquetes, CD-ROM, *pen-drive* e principalmente por disponibilização de conteúdo através da rede mundial de computadores.

De acordo com História (2014), a internet passou a ser comercializada no Brasil no ano de 1994 pela empresa de telecomunicação Embratel e, em 1995, passou a ser distribuída para a população, surgindo assim cursos on-line, web conferências e seminários. Hoje é possível fazer uma graduação ou pós-graduação através da internet, tudo regulamentado pela legislação brasileira:

- Decreto nº 5.622, de 19 de dezembro de 2005, que regulamenta o artigo 80 da Lei nº 9.394 de 20 de dezembro de 1996;
- Decreto nº 5.773, de 09 de maio de 2006, dispõe sobre o exercício das funções de regulação, supervisão e avaliação de instituições de educação superior e cursos superiores de graduação e sequenciais no sistema federal de ensino e;
- Decreto nº 6.303, de 12 de dezembro de 2007, altera dispositivos dos Decretos nos 5.622, de 19 de dezembro de 2005, que estabelece as diretrizes e bases da educação nacional, e 5.773, de 9 de maio de 2006, que dispõe sobre o exercício das funções de regulação, supervisão e avaliação de instituições de educação superior e cursos superiores de graduação e sequenciais no sistema federal de ensino.

Destaca-se a Associação Brasileira de Educação a Distância (ABED), fundada em 21 de junho de 1995, uma sociedade científica sem fins lucrativos. No endereço eletrônico <http://www.abed.org.br>, é possível encontrar artigos, bibliografias, links, legislações, vídeos, imagens, conferências, profissionais, instituições, enfim, uma grande quantidade de conteúdo para aprender sobre ou utilizar-se da educação a distância.

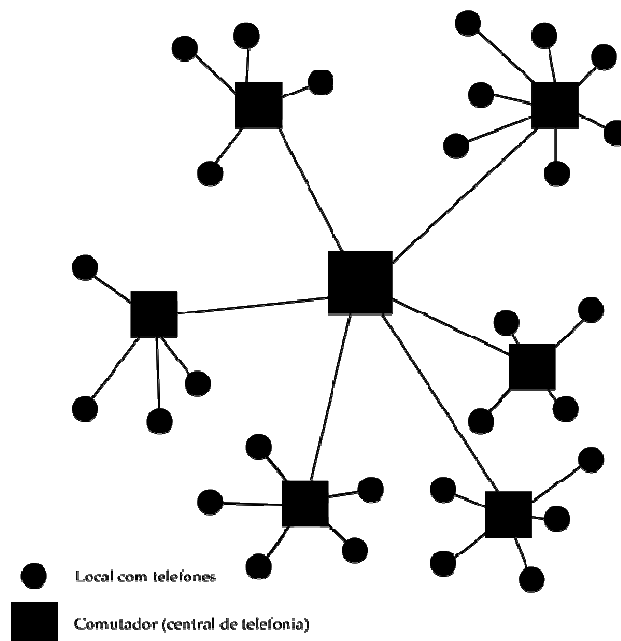
Visto a atual importância da internet para a educação a distância, faz-se necessário um estudo sobre sua história e evolução.

2.2 EVOLUÇÃO DA INTERNET

A internet, sendo um “[...] vasto conjunto de redes [...]” (TANENBAUM, 2003, p. 54) possibilita a conexão entre *hosts*³ em qualquer lugar do mundo. É possível participar de videoconferências, *chats*, conversas via VoIP⁴, assistir vídeos, ouvir músicas, compartilhar documentos, ensinar, aprender e trabalhar a qualquer hora com pessoas espalhadas pelo mundo. Pode-se, por exemplo, morar no Brasil e fazer um curso de especialização em uma grande universidade nos Estados Unidos ou trabalhar para uma empresa situada no Japão.

De acordo com Tanenbaum (2003, p. 54–59), a internet teve sua origem nos Estados Unidos na década de 1950, durante a guerra fria⁵. O Departamento de Defesa dos Estados Unidos, preocupava-se com uma guerra nuclear e com a quebra de comunicação devido a esta, já que toda a comunicação do departamento era feita por linha telefônica (figura 1) e era muito vulnerável, pois uma única central de comutação ligava diversos dispositivos e, se esta sofresse alguma avaria, deixaria milhares de telefones sem comunicação.

Figura 1: Estrutura do sistema de telefonia



Fonte: Tanenbaum (2003, p. 55)

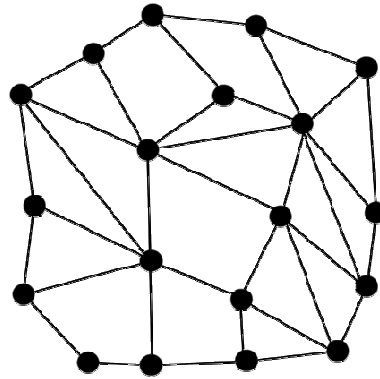
³ Um computador ou outro dispositivo em uma rede TCP/IP.

⁴ Abreviação de voz sobre IP. Utiliza-se a internet para a realização de conversas, como um telefone.

⁵ Período após a segunda guerra mundial (1945), que foi marcado por conflitos indiretos entre os Estados Unidos e a União Soviética.

Na década de 1960, Paul Baran desenvolveu um sistema digital de comutação de pacotes (figura 2), que encurtava a distância entre os dispositivos e conectava-os diretamente, fazendo com que o sistema fosse tolerante a falhas. Este sistema podia alcançar maior distância entre pontos de comutação sem sofrer interferências, o que não ocorria no sistema de telefonia convencional.

Figura 2: Sistema distribuído de comutação



Fonte: Tanenbaum (2003, p. 55)

Este sistema deu origem a internet que é conhecida hoje, mas ele não foi aceito pela empresa AT&T, que dominava o sistema de telefonia nos Estados Unidos, pelo simples motivo de não aceitarem que alguém tão jovem como Paul Baran lhes “ensinasse a criar um sistema telefônico”. A ideia de Paul Baran foi abandonada pelo Sistema de Defesa dos Estados Unidos e pelo próprio Baran, mas esta ideia é citada no ano de 1967 no Simpósio Sobre os Princípios do Sistema Operacional (SIGOPS - *Symposium on Operating System Principles*).

Larry Roberts, diretor da ARPA⁶, encontrou neste simpósio um documento que citava a pesquisa de Paul Baran e a sub-rede de comutação de pacotes, que ele mesmo chegou a desenvolver pesquisa. Este documento citava o projeto e implementação do sistema no campus do Laboratório Físico Nacional (NPL – *National Physical Laboratory*) sob a orientação de Donald Davies. Após este encontro, Larry Roberts decidiu levar adiante suas pesquisas e mais tarde, no ano de 1968, a sub-rede foi batizada de ARPANET e passou a ser estudada e desenvolvida por uma empresa de consultoria, chamada BBN.

⁶ *Advanced Research Projects Agency* – Uma organização de pesquisa de defesa, criada pelo presidente Eisenhower, para acabar com a disputa entre o Exército, Marinha e Força Aérea. Disputa que fez com que, em 1957, a União Soviética ultrapassasse os Estados Unidos na corrida espacial, com o lançamento do satélite artificial Sputnik. (ANDREW S. TANENBAUM, 2003, p. 55)

No ano de 1969, a ARPANET recebeu a colaboração de estudantes universitários do estado de Utah e, em poucos anos, passou a ser utilizada pela maioria das universidades dos Estados Unidos. Mais tarde, a ARPANET permitiu o envio de mensagens via satélite e também via rádio, a criação do Protocolo de Controle de Transmissão/Protocolo de Internet (TCP/IP – *Transmission Control Protocol/Internet Protocol*) e do Sistema de Nomeação de Domínio (DNS – *Domain Naming System*). Com o aumento da utilização, a ARPANET passou a ser procurada por muitas universidades, mas para utilizá-la, era necessário ter um contrato de pesquisa com o Departamento de Defesa dos Estados Unidos. Então, para driblar esta restrição e oferecer a ARPANET para qualquer grupo de pesquisa universitário, a Fundação Nacional de Ciências (NSF – *National Science Foundation*) criou a NSFNET no final da década de 1970.

Inicialmente, a NSFNET conectava seis supercomputadores da NSF, através do protocolo TCP/IP, em estados e cidades diferentes, criando assim a primeira Rede de Área Ampla (WAN - *Wide Area Network*)⁷. O sucesso da NSFNET fez com que, no ano de 1990, a NSF desvinculasse o seu sistema do governo Americano e buscasse parceria com a iniciativa privada, montando assim a Rede Avançada e Serviços (ANS – *Advanced Networks and Services*), que atualizou a NSFNET de 1,5 Mbps para 45 Mbps e formou a ANSNET. Em 1995, a ANSNET passou a ser comercializada, mas já era reconhecida como internet desde 1985.

No ano de 1989, o cientista Tim Bernes-Lee desenvolveu o que conhecemos como Rede Mundial de Computadores ou Web (WWW – *World Wide Web*), levando a internet para todos e agregando ao ambiente que somente era conhecido por pesquisadores e universitários, cultura e conhecimento de quase todo o planeta.

2.2.1 A INTERNET NO BRASIL

Após a criação da ARPANET no ano de 1968, de acordo com Carvalho (2006, p. 55), o Brasil pôde acessá-la no ano de 1975, durante o 8º Congresso Nacional de Processamento de Dados (CNPD), realizado pela Sociedade dos Profissionais e Usuários de TI⁸ do Estado de São Paulo (SUCESU - SP)⁹. Carvalho também apresenta o relato do presidente da SUCESU – SP:

⁷ É o tipo de rede que faz a comunicação entre dispositivos ou redes locais (LAN) em longas distâncias (cidades, países, continentes). (ANDREW S. TANENBAUM, 2003, p. 75)

⁸ Tecnologia da Informação: o conjunto de todas as atividades e soluções providas por recursos de computação que visam a produção, o armazenamento, a transmissão, o acesso, a segurança e o uso das informações.

Os terminais do Anhembi¹⁰ foram ligados via Embratel¹¹ aos terminais da rede americana, com demonstrações através de televisores e de um telão, para que a audiência pudesse acompanhar a conversa com os norte-americanos. Isso na época era uma grande novidade, tanto que convidamos o então Ministro das Comunicações, Quandt de Oliveira, para fazer a abertura do Congresso (SUCESU, 1987, p. 65 *apud* CARVALHO, 2006, p. 55).

Conforme Filippo e Sztajnberg (1996, p. 342–347), 5 anos após a apresentação da ARPANET aos brasileiros, no início da década de 1980 foi criado o Laboratório Nacional de Redes de Computadores (LARC) para que fosse possível gerenciar as pesquisas e criar a infraestrutura de redes no Brasil. Este processo demorou oito anos para dar resultados e, assim como nos anos de desenvolvimento e pesquisa da rede de computadores nos Estados Unidos da América, onde a ARPANET era utilizada no setor acadêmico e de pesquisas, no Brasil não foi diferente.

O Laboratório Nacional de Computação Científica (LNCC) do Rio de Janeiro, conectou-se a Bitnet¹² em setembro de 1988 por intermédio da Universidade de Maryland, localizada na cidade de College Park, seguido pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) em novembro de 1988 que conectou-se a Bitnet e a HEPnet¹³ através do Fermi National Laboratory, em Chicago e, por último, a Universidade do Rio de Janeiro (UFRJ) que conectou-se a Bitnet em maio de 1989, através da UCLA (Universidade da Califórnia) em Los Angeles. Este processo de criação da rede de computadores e conexão entre as instituições brasileiras e as instituições norte-americanas contou com as participações de Embratel, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Secretaria Especial de Informática (SEI). No ano de 1991, foi estabelecida uma conexão entre o LNCC e a FAPESP que, no ano citado, se tornaram polos de concentração, conectando também o Distrito Federal e os estados: Amazonas, Bahia, Mato Grosso do Sul, Minas

⁹ Fundada há 48 anos. Fomenta a evolução digital sustentável atraindo negócios entre os seus associados, através do relacionamento, conhecimento e parcerias, buscando a evolução contínua do conhecimento para o usuário de tecnologia de informação e comunicação no estado de São Paulo. Site: <http://www.sucesusp.org.br>.

¹⁰ Maior Centro de Eventos da América Latina, com espaços locais para realização de feiras de negócios, congressos, formaturas, shows e eventos esportivos.

¹¹ A Embratel é uma das maiores operadoras de telecomunicações do Brasil e faz parte do grupo América Móvil, líder no segmento na América Latina.

¹² Em 1981, a *City University of New York* (CUNY), após realizar uma pesquisa entre universidades, em um projeto liderado por Ira Fuchs (coordenador de informática), resolveu montar uma rede que pudesse interligar pessoas de uma forma simples e barata, criando a *Because It's Time Network* (BITNET) que inicialmente conectou a CUNY com a *Yale University*. (MARCELO SÁVIO REVOREDO MENEZES DE CARVALHO, 2006, p. 30)

¹³ A HEPNET (*High-Energy Physics Network*) era uma rede internacional baseada no protocolo de comunicação DECNET, que interligava instituições de estudo e pesquisa em física de altas energias. (MARCELO SÁVIO REVOREDO MENEZES DE CARVALHO, 2006, p. 86)

Gerais, Pará, Paraná, Pernambuco, Rio de Janeiro, Rio Grande do Norte, Rio Grande do Sul, São Paulo, e Sergipe. Esta nova rede permitia a comunicação entre estes pontos por meio de correio eletrônico operando a velocidade de 9600 bps¹⁴.

No ano de 1990, por iniciativa do Ministério de Ciência e Tecnologia (MCT) foi criada a Rede Nacional de Pesquisa (RNP), que contou com o apoio do SEI, CNPq, Financiadora de Estudos e Projetos (FINEP), Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), FAPESP e Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS). O trabalho deste grupo, seguindo o modelo do ano de 1970 da NSF, possibilitou a criação de um *backbone*¹⁵ de três níveis: nacional, regional e institucional. Este *backbone* foi apresentado no ano de 1992, na Conferência das Nações Unidas sobre Meio Ambiente e Desenvolvimento (ECO-92), realizada na cidade do Rio de Janeiro. Após esta apresentação, a RNP foi sendo aprimorada e passou a conectar-se via internet com velocidade de 64 Kbps (quilobit por segundo), o que possibilitou a comercialização deste serviço, inicialmente pela Telecomunicações do Estado do Rio de Janeiro (Telerj). Deste ponto em diante, a RNP passou por melhorias e expansão, permitindo que hoje, no Brasil, seja possível conectar à internet através da banda larga ou fibra ótica. De acordo com CONNECTIVITY (2017), a velocidade média alcançada ao fim do ano 2016 no Brasil é de 6391,7 Kbps, 99,87 vezes maior que a velocidade disponibilizada no ano de 1992 para comercialização.

Tanto trabalho e pesquisa empregados no passado com o intuito de conectar instituições de ensino e pesquisadores para aprender, aprimorar conhecimentos e desenvolver soluções, misturam-se com a realidade dos dias atuais, onde instituições de ensino buscam, através de ambientes virtuais de aprendizagem, ampliar a relação entre professor e aluno levando-a além do ambiente físico acadêmico e encorajando-os a mergulhar no mundo da informática e comunicação.

2.3 AMBIENTES VIRTUAIS DE APRENDIZAGEM

Ambiente Virtual de Aprendizagem (AVA) é um termo adaptado da língua inglesa: *Learning Management System* (LMS – Sistema de Gestão de Aprendizagem) e refere-se a um

¹⁴ bps: *bits per second* (bits por segundo). Velocidade máxima que pode ser atingida em transmissão de dados entre dispositivos conectados a uma rede. Quanto maior o valor, maior a velocidade de transmissão. Um bit corresponde a menor medida processável na informática e pode assumir dois valores: 0 (zero) ou 1 (um). O acrônimo *bit* significa: *Binary digiT* (dígito binário).

¹⁵ *Backbone* significa “espinha dorsal” e é o termo utilizado para identificar a rede principal pela qual os dados de todos os clientes da Internet passam. É a espinha dorsal da Internet. Site: <http://www.fundacaobradesco.org.br/vv-apostilas/IE8/O%20que%20%C3%A9%20Backbone.htm>.

programa ou sistema on-line, onde um professor cria cursos, disponibiliza aulas e compartilha conteúdo com seus alunos, que ingressam no ambiente do curso com autorização do professor. O termo AVA, apesar de não possuir uma definição oficial, pode ser definido por:

Um AVA está relacionado ao desenvolvimento de condições, estratégias e intervenções de aprendizagem num espaço virtual na web, organizado de tal forma que propicie a construção de conceitos, por meio da interação entre alunos, professores e objetos de conhecimento (VALENTINI; SOARES, 2005, p. 19 *apud* BELMONT; GROSSI, 2010, p. 4).

Na citação fica claro que somente ambientes específicos e desenvolvidos com tal propósito, podem ser classificados como ambiente virtual de aprendizagem, portanto, mesmo que possam ser utilizados com tal propósito, não são classificados como AVA os sites de compartilhamento de vídeos, redes sociais, fóruns, blogs, *wikis*¹⁶, etc. Existem diversos tipos de ambientes virtuais de aprendizagem disponíveis para utilização e, na sequência, apresentam-se três ambientes.

2.3.1 MOODLE

Moodle é um dos mais conhecidos ambientes virtuais de aprendizagem, é gratuito e não necessita de cadastro, basta fazer o download e, segundo Installation (2016), realizar a instalação em um servidor web Apache com linguagem de script PHP configurada e uma base de dados SQL. A palavra Moodle, segundo Common (2013) é um acrônimo que significa *Modular Object-Oriented Dynamic Learning Environment* (Ambiente de Aprendizagem Dinâmico Modular Orientado a Objeto), mas esta palavra pode ser encontrada em grandes dicionários da língua inglesa com o significado próximo a “*musings* (reflexão)”.

O sistema foi criado pelo desenvolvedor australiano Martin Dougiamas e, segundo History (2015), Martin lançou a versão oficial de nome Moodle 1.0 no ano de 2002, junto com o site <https://moodle.org>, que se tornou uma comunidade para as pessoas que estavam discutindo, criando temas e realizando traduções do Moodle para outros idiomas em outros sites, pudessem unir-se no site oficial. Isso porque o Moodle foi utilizado pela primeira vez no ano de 2001, por Martin e Peter Taylor na universidade de Curtin, através do site <http://moodle.com>, para demonstrar o andamento do projeto. Este primeiro site pode ser acessado pelo endereço <http://smec2001.moodle.com> e, na figura 3 pode-se conferir uma

¹⁶ *Wiki* é uma coleção de muitas páginas interligadas e cada uma delas pode ser visitada e editada por qualquer pessoa. Site: https://pt.wikipedia.org/wiki/Wikip%C3%A9dia:O_que_%C3%A9_uma_wiki.

captura de tela do primeiro post de Martin, disponibilizado por ele no fórum de discussão no site oficial.

Figura 3: Primeiro curso do Moodle

Fonte: <https://moodle.org/mod/forum/discuss.php?d=27533&parent=129848>

O Moodle tem código aberto (*Open Source*), isto significa que qualquer pessoa ou instituição tem acesso ao código do programa, podendo modificá-lo e utilizá-lo como bem entender, até mesmo desenvolver um novo programa e cobrar pelo desenvolvimento ou utilização.

“A licença não deve restringir qualquer parte de vender ou entregar o software como um componente de uma distribuição agregada de software que contém programas de várias fontes diferentes. A licença não exigirá royalty ou outra taxa para tal venda¹⁷” (OPEN, [1998?]).

Gratuidade e customização são fatores relevantes para que o Moodle seja utilizado em 236 países e tenha 81239 sites registrados em sua base de dados. O Brasil é o terceiro país em quantidade de registros, totalizando 4581 sites (Cf.: MOODLE S., [2017?]). Mas não é só isso; ele também oferece ferramentas importantes para alunos e professores além da

¹⁷ The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale. (OPEN SOURCE INITIATIVE, [1998]. *Tradução Nossa*)

disponibilização e acesso ao conteúdo de aula, o professor pode criar tarefas com tempo de entrega e tipo de conteúdo anexo que deverá ser entregue pelo aluno, pode criar calendários, disponibilizar aulas gratuitas ou pagas (depende de instalação de serviço de terceiros) e restringir o acesso a aulas e conteúdo. Existe também o serviço de chat e fórum de discussão acessível aos alunos.

Na página <https://download.moodle.org> está disponível para download a versão 3.3+ lançada em 24 de maio de 2017 e também versões anteriores, observando-se que cada versão do Moodle exige versões específicas de linguagem PHP e do SGBD. Os requisitos necessários para cada versão estão descritos junto ao link de download de cada versão. O Moodle também oferece pacotes de idiomas, sua língua nativa é o inglês, mas pela cooperação de usuários, ele oferece tradução para diversos idiomas, incluindo a língua portuguesa.

O processo de instalação do Moodle depende de conhecimentos específicos. Inicialmente, deve-se acessar o SGBD no ambiente de hospedagem web, criar uma base de dados, definir usuário, senha e permissões de acesso para o usuário criado. Em seguida, após fazer o download do arquivo que contém o pacote de instalação, deve-se descompactá-lo e proceder o upload das pastas e arquivos ao ambiente de hospedagem por meio do FTP. O próximo passo é abrir o navegador web e digitar na barra de endereços o domínio acrescentando-se “/” (barra) e o nome da pasta onde foi realizado o upload do pacote de instalação. Exemplo: o domínio do site é “meusite.com.br” e a pasta “moodle”, então deve-se digitar: meusite.com.br/moodle. Surge na tela do navegador a página inicial de instalação do Moodle, onde escolhe-se o idioma que será utilizado no ambiente, seguindo, escolhe-se o driver do banco de dados e insere-se as informações do mesmo, como: *host*¹⁸ da base de dados, nome da base de dados, nome de usuário e senha. Na sequência, o instalador verifica as condições do ambiente de hospedagem e exibe um relatório com a possibilidade de proceder a instalação e, caso encontre alguma pendência, exibe as informações sobre procedimentos a serem executados. Após essa verificação é realizada a instalação, que ocorre em alguns minutos.

Finalizando a instalação, deve-se configurar a utilização do Moodle, criando um usuário administrador, inserindo informações como país e cidade, configurar as informações da página inicial e o modo de autenticação da conta de novos usuários.

¹⁸ Computador principal, ligado permanentemente à rede, que comanda as operações de outros computadores nessa rede e permite o acesso de usuários. (Michaelis - Dicionário da Língua Portuguesa)

2.3.2 TELEDUC

TelEduc é um ambiente virtual de aprendizagem de código aberto, com lançamento oficial no ano de 2001. Seu projeto teve início no ano de 1997 com a dissertação de mestrado de Alessandra de Dutra e Cerceau, orientação da professora Dra. Heloísa Vieira da Rocha, realizada na Universidade Estadual de Campinas (UNICAMP), pelo Núcleo de Informática Aplicada à Educação (NIED) e parceria com o Instituto de Computação (IC) (Cf.: HISTÓRICO. [2017?]).

[...] Ainda no ano de 2002, o Projeto TelEduc foi premiado pela Associação Brasileira de Educação a Distância (ABED) na categoria Pesquisa e Desenvolvimento em EaD, obtendo o 1º lugar nacional.

A partir de 2003 o TelEduc passou a ser usado pelo Projeto Ensino Aberto da Pró-Reitoria de Graduação da UNICAMP que tem por objetivo oferecer um ambiente computacional de apoio às disciplinas presenciais do catálogo de cursos da Universidade. O Conselho Universitário da UNICAMP (CONSU), parabenizou a equipe da Profa. Heloísa pela iniciativa no desenvolvimento do software.

[...] Esses resultados são fruto do trabalho e da dedicação de uma equipe formada [por] estagiários e bolsistas SAE¹⁹, alunos de graduação em Ciência da Computação e áreas afins. Ao longo desses vários anos de vida do Projeto tivemos a oportunidade de desenvolver o primeiro ambiente de ensino-aprendizagem para a Internet no Brasil e de formar dezenas de profissionais capacitados para atuar na área de desenvolvimento de tecnologia para a Educação (HISTÓRICO. [2017?]).

TelEduc é um sistema que conta com a participação de muitas pessoas e auxilia no desenvolvimento profissional de estudantes da UNICAMP que contribuem para a manutenção e aprimoramento deste. É também o primeiro ambiente virtual de aprendizagem desenvolvido no Brasil e, apesar de estar disponível em 4 idiomas, português (Brasil), português (Portugal) inglês e espanhol, de acordo com Quem ([2017?]), só existe relato de utilização no Brasil, em 14 estados e no Distrito Federal, totalizando 56 instituições.

O sistema está em sua versão 4.4.0 e para utilizá-lo, é necessário realizar o download do pacote de instalação na página <http://www.teleduc.org.br/?q=downloads> do site do TelEduc. Nesta página, além do pacote de instalação, fica disponível o guia de instalação. O sistema deve ser instalado em um servidor web Apache com linguagem de script PHP configurada e base de dados MySQL, devendo atentar-se para a versão correta destes conforme descrição no guia de instalação. O NIED oferece de forma gratuita a criação de cursos TelEduc em seu servidor, sendo estes cursos sem fins lucrativos e necessitando de

¹⁹ Serviço de Apoio ao Estudante da UNICAMP. Site: <https://www.portal.sae.unicamp.br>.

aprovação pela equipe do TelEduc. As informações necessárias e e-mail de contato estão disponíveis no endereço <http://www.teleduc.org.br/?q=criar-curso>.

TelEduc disponibiliza a seus usuários, além do compartilhamento de conteúdo de cursos: agenda, tarefas, avaliação, correio, bate papo, fórum de discussão, mural de informações, enquete, portfólio e outras. Através do endereço <http://www.teleduc.org.br/?q=visao-geral-ferramentas> é possível visualizar todas as ferramentas disponíveis e suas descrições.

A instalação do TelEduc pode ser realizada da mesma maneira que a instalação do Moodle, cria-se a base de dados, descompacta-se e faz-se o upload do pacote de instalação ao ambiente de hospedagem web e acesso ao instalador através do navegador web. A diferença está na necessidade de criar uma pasta com o nome “arquivos” dentro da pasta do pacote de instalação e definir permissões de leitura, gravação e execução para todos os usuários.

2.3.3 EDMODO

Edmodo é um ambiente virtual de aprendizagem com bastante recursos e de fácil utilização. Não é necessário possuir um ambiente de hospedagem web, conhecimentos técnicos e nem pagar por sua utilização. Acessando o endereço <https://www.edmodo.com>, basta selecionar o perfil de professor e criar uma conta através de seu e-mail ou utilizar sua conta do Google²⁰ ou do Office 365²¹ (Cf.: EDMODO, 2016).

Edmodo tem características e funcionamento de um ambiente virtual de aprendizagem, mas de acordo com Sobre (2016), “O Edmodo é uma rede global de educação que ajuda a conectar todos os alunos às pessoas e recursos necessários para o atingimento do potencial máximo.”, ou ainda “[...] uma rede social educacional número um para aluno do ensino fundamental ou ensino médio de todo o mundo [...]” (SOBRE, 2016). A empresa foi fundada no ano de 2008 na cidade de Chicago, Illinois – EUA por Nic Borg, Jeff O'Hara e Crystal Hutter, atualmente está situada na cidade de San Mateo, California – EUA. Possui 78.973.087 usuários e o site está disponível em diversos idiomas, incluindo a língua portuguesa, mas com algumas traduções incompletas, fazendo com que o parte do texto

²⁰ Google Inc. é uma empresa multinacional de serviços online e software dos Estados Unidos. O Google hospeda e desenvolve uma série de serviços e produtos baseados na internet e gera lucro principalmente através da publicidade.

²¹ Office 365 é uma suíte de aplicativos para escritório online por assinatura que oferece acesso a vários serviços e softwares construídos em torno da plataforma Microsoft Office.

apareça em língua inglesa. Já o ambiente de aprendizagem está na língua inglesa, sem opções de traduções disponíveis.

Sua utilização é simples, basta criar um grupo com o nome do curso, enviar o código ou link de inscrição, que é gerado automaticamente, para os alunos e começar a postar textos ou arquivos anexos através do link de upload ou através da integração com Google Drive²² ou One Drive²³. É possível criar tarefas e corrigi-las, criar calendários e questionários. As aulas do curso são postagens, onde pode-se realizar comentários e estabelecer conversas com os participantes do grupo. É possível que pais de alunos criem contas no Edmodo somente para acompanhar a evolução de seus filhos no curso e, estas contas são gerenciadas pelo professor.

Existe uma grande quantidade de ambientes virtuais de aprendizagem à disposição com nomes, modelos, formas de utilização, ferramentas disponíveis, integração ou não com outros serviços e preços diferentes que, de maneira geral, buscam auxiliar professores a compartilhar conhecimento e facilitar aos alunos o acesso as informações compartilhadas.

Estes ambientes são desenvolvidos, em sua maioria, utilizando-se da programação orientada a objetos, seja ela PHP, Java, C# (CSharp) ou qualquer outra da preferência ou necessidade do desenvolvedor, que auxilia na criação de módulos (pedaços de programa) que podem ser modificados individualmente ou incluídos novos módulos, sem a necessidade de alterar toda a estrutura do programa. Visto que a programação orientada a objetos é parte importante de um ambiente virtual de aprendizagem e também, a forma de programação mais utilizada na atualidade, se faz necessário um estudo sobre o tema.

2.4 PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação orientada a objetos (POO) é um padrão de desenvolvimento utilizado por algumas linguagens de programação, entre elas, Java e C#. Este padrão descreve o mundo como um conjunto de objetos que interagem entre si e, cada objeto, possui características (atributos) e padrões de comportamento (métodos) que, tanto atributos quanto métodos, podem ser visíveis (públicos) ou invisíveis (privados) para outros objetos. Estes objetos podem possuir herança, assim como um filho pode herdar a cor dos cabelos (característica) ou torcer pelo mesmo time de futebol (comportamento) de seu pai ou avô, os objetos também podem herdar atributos e métodos. Todo objeto é proveniente de uma classe, que pode ser comparada a uma família, exemplo: Maria, João, Antonieta e José são membros (instâncias)

²² Serviço de armazenamento, sincronização e compartilhamento de arquivos em nuvem do Google.

²³ Serviço de armazenamento, sincronização e compartilhamento de arquivos em nuvem da Microsoft.

da família Silva e, podem herdar características de seus antecessores como, a cor da pele, olhos e cabelos, o formato do rosto, doenças hereditárias e outras. Mas os membros de uma família também possuem características próprias como, o gosto musical, carreira profissional, características físicas, etc.

De acordo com Ricarte (2001, p. 3), “objetos são instâncias de classes, que determinam qual informação um objeto contém e como ele pode manipulá-la”, e classe é “[...] um gabarito para definição de um objeto”, sendo responsável por descrever os atributos e métodos de um objeto.

Utilizando como exemplo um ambiente virtual de aprendizagem, pode-se criar uma classe nomeada como **Pessoa**, a qual servirá de gabarito para a definição de professores e alunos. Na classe Pessoa, declara-se os atributos **nascimento** e **nome** e implementa-se um método para receber estes atributos. Perceba que a classe Pessoa, utilizada neste exemplo, será utilizada em um programa que é um ambiente virtual de aprendizagem, mas esta mesma classe pode ser reutilizada em qualquer outro programa que necessite de uma pessoa com os atributos nascimento e nome, por exemplo, um programa para consultório médico ou de uma loja. Esta é uma das grandes vantagens da programação orientada a objetos, o reaproveitamento ou reutilização de código.

Visto que uma classe é um gabarito para definir um objeto, Ricarte (2001, p. 5) diz que “um objeto é um elemento que representa, no domínio da solução, alguma entidade (abstrata ou concreta) do domínio de interesse do problema sob análise”. No exemplo anterior, a classe Pessoa dará origem a objetos, neste caso, entidades concretas que poderão ser a professora Maria, o professor João, a aluna Jéssica, o aluno Pedro, etc.

No paradigma de orientação a objetos, tudo pode ser potencialmente representado como um objeto. Sob o ponto de vista da programação orientada a objetos, um objeto não é muito diferente de uma variável normal. Por exemplo, quando define-se uma variável do tipo **int** em uma linguagem de programação como C ou Java, essa variável tem: um espaço em memória para registrar o seu estado (valor); um conjunto de operações que podem ser aplicadas a ela, através dos operadores definidos na linguagem que podem ser aplicados a valores inteiros. Da mesma forma, quando se cria um objeto, esse objeto adquire um espaço em memória para armazenar seu estado (os valores de seu conjunto de atributos, definidos pela classe) e um conjunto de operações que podem ser aplicadas ao objeto (o conjunto de métodos definidos pela classe) (RICARTE, 2001, p. 5, grifo do autor).

Os objetos, apesar de serem mais complexos podem ser comparados a uma variável, que são armazenadas em um espaço da memória volátil²⁴ do computador, sendo descartados logo após sua utilização.

A programação orientada a objetos, visando garantir a integridade de dados, possui o recurso denominado encapsulamento, que é uma forma de tornar privados atributos de uma classe, impedindo que fiquem visíveis à outras classes. Conforme Ricarte (2001, p. 5), encapsulamento “[...] é o princípio de projeto pelo qual cada componente de um programa deve agregar toda a informação relevante para sua manipulação como uma unidade (uma cápsula) [...]”.

²⁴ Memória RAM (*Random Access Memory* – Memória de Acesso Aleatório). É a memória temporária do computador, onde são armazenados dados que um programa necessita para executar uma tarefa. Estes dados são descartados automaticamente logo após sua utilização.

A representação aparece em uma parte da especificação chamada de privada, introduzida pela palavra reservada *private*. A cláusula privada sempre aparece no final do pacote de especificação. Ela é visível para o compilador, mas não para as unidades de programa cliente (SEBESTA, 2010, p. 510).

A palavra reservada *private* é a cláusula ou definição da visibilidade do atributo, também conhecido por modificador de acesso, declarada antes da declaração do tipo²⁵ do atributo, seguido pelo nome, que é definido pelo programador. Um atributo privado não é visível por outras classes, mas pode ser acessado, em outras classes, por intermédio de um método com cláusula *public*, que é visível para todas as classes do programa, sendo este, responsável por retornar (pegar) ou alterar o valor do atributo diretamente ou, indiretamente através de seu algoritmo. Ainda existe a cláusula *package*, que permite que apenas as classes que estão no mesmo pacote²⁶ tenham acesso ao conteúdo e a cláusula *protected*, que permite que subclasses acessem o conteúdo. Nas linguagens de programação Java e C#, é possível criar métodos assessores, que são específicos para retornar ou definir o valor de um atributo.

Em Java, estes métodos são conhecidos como *getters* para retornar o valor do atributo e, *setters* para definir o valor do atributo. Acompanhe o exemplo:

Definição do atributo **nota** com cláusula privada e do tipo inteiro. Este atributo é visível apenas em sua classe, a classe **Aluno**.

```
private int nota = 0
```

Método de cláusula pública, que retorna um número inteiro e foi nomeado pelo programador como **getNota** e retorna o valor do atributo **nota** definido na classe. Este método é chamado de *getter* e é um bom procedimento de programação, iniciar o nome do método com a palavra *get*.

```
public int getNota() {
    return nota;
}
```

Método de cláusula pública, sem retorno de valor (*void*), nomeado pelo programador como **setNota**, e define um novo valor para o atributo **nota**. Este método é chamado de *setter* e é um bom procedimento de programação começar o nome do método pela palavra *set*.

²⁵ O tipo do atributo especifica se o atributo é um caractere (*char*), uma coleção ou *array* de caracteres (texto ou *string*), um número inteiro (*int*), um número real (*double* ou *float*). Estes são os tipos mais comuns em linguagem de programação, podendo receber notações diferentes.

²⁶ Diretório ou pasta onde ficam armazenadas as classes do programa.

```
public void setNota(int novaNota) {
    this.nota = novaNota;
}
```

Criação da instância da classe **Aluno**, o objeto **aluno**, em outra classe. Implementação de um método para definir a nota do aluno e exibi-la junto com a informação de aprovação ou reprovação.

```
Aluno aluno = new Aluno();
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Digite a nota do aluno: ");
    aluno.setNota(sc.nextInt());
    if(aluno.getNota >= 6)
        System.out.println("Parabéns! Você passou!");
    else
        System.out.println("Reprovado!");
}
```

Em C# existe apenas um método, definido como propriedade, que possui dois blocos de acesso, *get* e *set*. Acompanhe o exemplo:

Definição do atributo **nota** com cláusula privada e do tipo inteiro. Este atributo é visível apenas em sua classe, a classe **Aluno**.

```
private int nota = 0
```

Definição da propriedade do atributo **nota**, de cláusula pública. Pode retornar (*get*) ou definir (*set*) o valor do atributo **nota**. Esta propriedade foi nomeada como **Nota** e é um bom procedimento de programação que a propriedade tenha o mesmo nome do atributo com a letra inicial maiúscula.

```
public int Nota
{
    get { return nota; }
    set { nota = value; }
}
```

Criação da instância da classe **Aluno**, o objeto **aluno**, em outra classe. Implementação de um método para definir a nota do aluno e exibi-la junto com a informação de aprovação ou reprovação.

```

Aluno aluno = new Aluno();
static void Main(string[] args)
{
    Console.WriteLine("Digite a nota do aluno: ");
    aluno.Nota = int.Parse(Console.ReadLine());
    if(aluno.Nota >= 6)
        Console.WriteLine("Parabéns! Você passou!");
    else
        Console.WriteLine("Reprovado!");
}

```

Nos exemplos de encapsulamento descritos anteriormente, a nota somente pode ser manipulada criando-se o objeto **aluno**, que é uma instância da classe **Aluno**, e acessando o atributo através dos métodos *get* e *set* em Java ou através da propriedade em C#.

“Uma linguagem orientada a objetos deve fornecer suporte para três recursos chave de linguagem: tipos de dados abstratos, herança e vinculação dinâmica de chamadas a métodos” (SEBESTA, 2010, p. 547).

O ocultamento de informação presente no encapsulamento, é considerado uma forma de abstração, onde permite-se esconder atributos não significativos para todas as classes do programa. A abstração também diz respeito a atributos comuns a todos os objetos provenientes de uma classe, de acordo com Sebesta (2010, p. 504):

[...] abstração é uma visão ou representação de uma entidade que inclui apenas os atributos mais significativos. De um modo geral, a abstração permite que alguém colete exemplares de entidades em grupos nos quais seus atributos comuns não precisam ser considerados. Por exemplo, suponha que definíssemos aves como criaturas com os seguintes atributos: duas asas, duas pernas, um rabo e penas. Então, se dissermos que um corvo é uma ave, uma descrição de um corvo não precisa incluir esses atributos. O mesmo ocorre para os pisco-de-peito-ruivo, pardais e pica-paus de barriga amarela. Esses atributos comuns nas descrições de espécies específicas de pássaros podem ser abstraídos.

Dentro de uma espécie em particular, apenas os atributos que a distinguem precisam ser considerados, resultando em uma simplificação significativa das descrições. Uma visão menos abstrata de uma espécie, aquela de um pássaro, pode ser considerada quando for necessário ver um alto nível de detalhes. No mundo das linguagens de programação, a abstração é uma arma contra a complexidade da programação; seu propósito é simplificar o processo de programação. É uma arma efetiva, pois permite que os programadores foquem em atributos essenciais, enquanto ignoram os subordinados.

O conceito de abstração descrito por Sebesta remete ao recurso de herança, presente na programação orientada a objetos. A herança diz respeito a uma classe “pai”, que contém atributos e métodos comuns a todas as classes “filhas” que, por sua vez, poderão agregar

atributos e métodos específicos. Voltando ao exemplo do ambiente virtual de aprendizagem, este deverá possuir uma classe Aluno e uma classe Professor, ambas as classes possuem atributos comuns, como nome e data de nascimento. A classe Aluno, por sua vez, deverá conter o atributo nota, enquanto a classe Professor não necessita deste atributo. Então pode-se criar uma classe nomeada como Pessoa que deverá conter os atributos nome e data de nascimento, sendo que as classes Aluno e Professor deverão herdar estes atributos. Acompanhe o exemplo em Java:

Classe Pessoa, com dois atributos e um método construtor (método utilizado em outra classe para criar o objeto) tem como parâmetros o nome e a data de nascimento.

```
public class Pessoa {
    private String nome;
    private String nascimento;
    public Pessoa(String nome, String nascimento){
        this.nome = nome;
        this.nascimento = nascimento;
    }
}
```

Classe Aluno, que herda (*extends*) a classe Pessoa, tem o atributo nota e um método construtor que recebe como parâmetro o atributo nota e os atributos da classe pai – *super(nome, nascimento)*.

```
public class Aluno extends Pessoa {
    private int nota;
    public Aluno(String nome, String nascimento, int nota){
        super(nome, nascimento);
        this.nota = nota;
    }
}
```

Classe Professor, que herda (*extends*) a classe Pessoa e tem um método construtor que recebe como parâmetros os atributos da classe pai – *super(nome, nascimento)*.

```
public class Professor extends Pessoa {
    public Professor(String nome, String nascimento){
        super(nome, nascimento);
    }
}
```

Em C#, a herança é um pouco diferente de Java, mas segue o mesmo princípio. Acompanhe o exemplo:

Classe Pessoa, com duas propriedades de atributo. Não se implementa método construtor.

```
class Pessoa {
    public string Nome { get; set; }
    public string Nascimento { get; set; }
}
```

Classe Aluno, que herda (:) a classe Pessoa, tem o atributo nota e método construtor com parâmetros que recebem atributos da classe pai.

```
class Aluno : Pessoa {
    private int nota;
    public Pessoa(string nome, string nascimento, int nota)
    {
        Nome = nome;
        Nascimento = nascimento;
        this.nota = nota;
    }
}
```

Classe Professor, que herda (:) a classe Pessoa e possui método construtor com parâmetros que recebem atributos da classe pai.

```
class Professor : Pessoa {
    public Pessoa(string nome, string nascimento)
    {
        Nome = nome;
        Nascimento = nascimento;
    }
}
```

A criação do objeto para utilização em outra classe, tanto em Java quanto em C#, segue o mesmo procedimento. Acompanhe o exemplo:

```
public class NovaClasse {
    Aluno james = new Aluno("James Gosling", "19/05/1955", 10);
    Aluno anders = new Aluno("Anders Hejlsberg", "02/12/1960", 9);
    Professor alan = new Professor("Alan Curtis Kay",
    "17/05/1940");
}
```

A classe Aluno, independentemente da existência de herança, deu origem a dois objetos, **james** e **anders**, enquanto a classe Professor, neste exemplo, gerou apenas o objeto **alan** e, nos três casos, houve a herança dos atributos definidos pela classe pai, a classe Pessoa.

A herança não ocorre apenas através do método construtor, todo objeto filho herda os atributos e métodos da classe pai, podendo invoca-los diretamente, por exemplo em C#:

Classe Aluno, apenas com a propriedade de atributo nota.

```
public class Aluno
{
    public Nota { get; set; }
}
```

Criação do objeto aluno em outra classe.

```
Aluno aluno = new Aluno();
aluno.Nome = "Joaquim";
aluno.Nascimento = "01/01/1901";
aluno.Nota = 10;
```

O objeto aluno foi criado através do construtor padrão da classe, o construtor vazio, que não recebe argumentos. Os atributos Nome e Nascimento oriundos da classe pai são invocados diretamente pelo objeto da classe filha, o que agiliza o desenvolvimento do programa, mas por outro lado, se o método construtor exige a passagem de argumentos, isso garante que não existirá nenhum aluno sem nome ou sem data de nascimento, o que pode ocorrer se o construtor não tem parâmetros em sua declaração. Sebasta (2010, p. 547) especifica claramente a importância da herança na programação orientada a objetos.

A herança oferece uma solução tanto para o problema de modificação oriundo do reuso de tipos abstratos de dados quanto para o de organização de programas. Se um novo tipo abstrato de dados pode herdar os dados e funcionalidades de algum tipo existente, e também é permitido que ele modifique algumas das entidades e adicione novas, o reuso é amplamente facilitado sem requerer mudanças ao tipo abstrato de dados reutilizado. Os programadores podem começar com um tipo abstrato de dados existente e projetar um descendente modificado dele para atender a um novo requisito do problema. Além disso, a herança fornece um framework para a definição de hierarquias de classes relacionadas que pode refletir os relacionamentos de descendência no espaço do problema.

A abstração de dados, da qual a herança é parte fundamental, auxilia o programador no desenvolvimento de soluções para problemas, visto que o reuso de atributos e métodos agiliza no tempo de desenvolvimento e pode exigir que todos os membros da hierarquia de classes possuam propriedades e comportamentos comuns a todos e possam apresentar características únicas.

A herança não se restringe apenas a uma classe pai e uma classe filha (herança simples), uma classe filha pode ser classe pai de outra classe (herança múltipla). Portanto, esta

terceira classe herdará os atributos e métodos da segunda e da primeira classe (Cf.: SEBESTA, 2010, p. 548).

A nomenclatura dos componentes da programação orientada a objetos, segundo Sebesta (2010, p. 547–548) provém da linguagem precursora, Simula 67, criada entre 1962 e 1968 por Kristen Nygaard e Ole-Johan Dal, lembrando que a orientação a objetos somente ganhou reconhecimento no ano de 1980 com a criação da linguagem Smalltalk 80, desenvolvida por Alan Curtis Kay. Veja a nomenclatura de componentes descrita por Sebesta:

- Classe: tipo abstrato de dado;
- Objeto: instância de classe;
- Classe derivada ou subclasse: uma classe definida por meio de herança de outra classe;
- Classe pai ou superclasse: uma classe da qual a nova é derivada;
- Métodos: subprogramas que definem as operações em objetos de uma classe.
- Mensagens: chamadas de métodos e;
- Protocolo de mensagens ou interface de mensagens: a coleção completa de métodos de um objeto.

Em programação orientada a objetos, existe o recurso de polimorfismo, que é um recurso utilizado dentro do contexto de herança.

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse (RICARTE, 2001, p. 6).

Ricarte define o polimorfismo de forma didática, através da utilização de métodos com mesma assinatura²⁷ e, Sebesta (2010, p. 552), apresenta uma definição técnica que diz que o polimorfismo:

[...] é definido como o uso de um ponteiro ou referência polimórfica para acessar um método cujo nome é sobrescrito na hierarquia de classes que define o objeto para ao qual o ponteiro ou referência apontam. A variável polimórfica é o tipo da classe base, e esta define ao menos o protocolo de um método que é sobrescrito pelas classes derivadas. A variável polimórfica pode referenciar objetos da classe base e de classes descendentes, então a classe do objeto para o qual ela aponta nem sempre pode ser determinada estaticamente. A vinculação de mensagens a métodos enviadas por variáveis polimórficas deve ser dinâmica. [...]

²⁷ Acesso, retorno, nome e parâmetros. Ex: *public int calcularIdade(Date nascimento)*, onde: *public* é o acesso, *int* é o retorno, *calcularIdade* é o nome do método e *(Date nascimento)* é o parâmetro a ser recebido.

Ambas as definições se referem ao fato de, quando houver herança entre classes, a subclasse pode fazer alterações em um método para que ele se adapte a necessidade do objeto e, essa alteração recebe o nome de sobrescrita. Como exemplo, demonstra-se o cálculo de bonificação para funcionários de uma empresa, onde estes têm uma bonificação de 10% sobre o salário mensal, enquanto à bonificação dos gerentes adiciona-se o percentual de vendas.

Exemplo em Java:

Superclasse **Funcionario**. O método retorna um número real, que corresponde ao valor da bonificação e o atributo **salario** tem a cláusula de acesso definida como *protected*, assim a classe filha poderá utilizá-la.

```
public class Funcionario {
    protected double salario;
    public double calcularBonificacao() {
        return salario * 0.10;
    }
}
```

Subclasse **Gerente**. Esta classe tem o atributo privado **percentualVendas**, que é exclusivo dos gerentes. A sobrescrita do método **calcularBonificacao** retorna o cálculo de bonificação comum a todos os funcionários adicionado ao valor exclusivo da bonificação do gerente. A instrução **super.calcularBonificacao()** invoca o método declarado na superclasse, mantendo assim os 10% de bonificação e, caso o valor da bonificação dos funcionários seja alterado, a subclasse recebe esta modificação sem intervenção do programador. A anotação **@Override** é necessária para garantir a sobrescrita do método, assim as alterações no método da superclasse aplicam-se ao método da subclasse.

```
public class Gerente extends Funcionario {
    private double percentualVendas;
    @Override
    public double calcularBonificacao() {
        return super.calcularBonificacao() +
            (salario * percentualVendas);
    }
}
```

Em C#, seguem-se os mesmos conceitos, aplicando-se as diferenças da linguagem. Acompanhe o exemplo:

Superclasse **Funcionario**. Utiliza-se a palavra-chave **virtual** para permitir a sobrescrita do método na subclasse.

```

class Funcionario
{
    protected double salario;
    public virtual double calcularBonificacao()
    {
        return salario * 0.10;
    }
}

```

Subclasse **Gerente**. A instrução **base.calcularBonificacao()** invoca o método declarado na superclasse, mantendo assim os 10% de bonificação e, caso o valor da bonificação dos funcionários seja alterado, a subclasse recebe esta modificação sem intervenção do programador. A palavra-chave **override** é necessária para garantir a sobrescrita do método, assim as alterações no método da superclasse aplicam-se ao método da subclasse.

```

class Gerente : Funcionario
{
    private double percentualVendas;
    public override double calcularBonificacao()
    {
        return base.calcularBonificacao() +
            (salario * percentualVendas);
    }
}

```

Utilizar o polimorfismo auxilia o programador na implementação do algoritmo do método, agilizando os processos de desenvolvimento e manutenção do programa, sendo que uma modificação no método da superclasse reflete diretamente no método de suas subclasses.

Sebesta (2010, p. 547) diz que “uma linguagem orientada a objetos deve fornecer suporte para três recursos chave de linguagem: tipos de dados abstratos, herança e vinculação dinâmica de chamadas a métodos” e, esses recursos chaves formam os quatros pilares da programação orientada a objetos, apresentados nesta seção: abstração, encapsulamento, herança e polimorfismo.

2.4.1 JAVA

Java é uma linguagem de programação orientada a objetos baseada na linguagem de programação C++, criada em 1990 por James Gosling, programador, *fellow* e vice-presidente

da Sun Microsystems²⁸. Java teve início através do *Project Green*, que recebeu este nome porque a porta do escritório onde estava sendo desenvolvido o projeto era da cor verde. A linguagem de programação originada deste projeto chamava-se *Oak* (carvalho), porque James Gosling olhou pela janela do escritório e avistou um carvalho, sendo alterada para Java por questões de conflito de marca (Cf.: SEBESTA, 2010, p. 660).

De acordo com o site da Oracle, atual proprietária do Java:

Java é uma linguagem de programação e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995. Existem muitas aplicações e sites que não funcionarão, a menos que você tenha o Java instalado, e mais desses são criados todos os dias. O Java é rápido, seguro e confiável. De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares (O QUE, [2017?!]

Apesar do uso atual da linguagem Java nos dispositivos mencionados, de acordo com Sebesta (2010, p. 114–117) ela foi desenvolvida para utilização em dispositivos eletrônicos embarcados, como aparelhos de TV, geladeiras e micro-ondas. Entretanto, nunca comercializaram os dispositivos que utilizavam Java e, ela somente passou a ser utilizada no ano de 1993 com a chegada da rede mundial de computadores e navegadores gráficos, no desenvolvimento de aplicações web e principalmente de pequenos programas chamados de *applets*, que eram incluídos e exibidos nos navegadores.

Uma aplicação executando Java era dez vezes mais lenta que outra compilada em linguagem C, isto devido a seu interpretador, uma máquina virtual denominada JVM (Java *Virtual Machine* – Máquina Virtual Java). Este interpretador oferece portabilidade ao programa, que pode ser executado em qualquer sistema operacional precedendo-se a instalação do JRE (Java *Runtime Environment* – Ambiente de Execução Java) adequado. O JRE possui a JVM e as classes de núcleo da plataforma Java que, atualmente possibilitam a compilação do código em tempo de uso através de um compilador JIT (*Just-in-Time*). O JIT aumentou a velocidade e eficiência de programas Java, tornando a linguagem competitiva com outras. A compilação de um programa Java gera o *bytecode*, que é um código intermediário interpretado pela JVM que interage diretamente com o sistema operacional.

²⁸ Sun Microsystems foi adquirida pela Oracle no ano de 2010. Mais informações no endereço eletrônico: <https://www.oracle.com/sun/index.html>.

Java é uma linguagem de programação simples e segura. É totalmente gratuita e possui ambientes de desenvolvimento gratuitos e pagos. Eclipse é um IDE²⁹ gratuito, multiplataforma e desenvolvido pela Eclipse *Foundation*. A Oracle oferece dois ambientes de desenvolvimento gratuitos: o Netbeans, muito funcional com relação a construção de ambiente gráfico, oferece suporte a banco de dados e *plug-ins*³⁰ de terceiros, e o JDeveloper, que oferece integração com plataformas e serviços da Oracle.

2.4.2 C#

C# (lê-se “cê sharpe”) é uma linguagem de programação orientada a objetos, baseada em C++ e Java, desenvolvida no ano 2000 por uma equipe da Microsoft que tinha como projetista líder o engenheiro de software, de nacionalidade dinamarquesa, Anders Hejlsberg (Cf.: SEBESTA, 2010, p. 125).

Conforme definição no site Microsoft *Developer Network*:

C# é uma linguagem orientada a objeto elegante e de tipo seguro que permite aos desenvolvedores criar uma variedade de aplicativos seguros e robustos que são executados no .NET *Framework*³¹. É possível usar C# para criar muito aplicativos de cliente do Windows, serviços Web XML, componentes distribuídos, aplicativos de cliente-servidor, aplicativos de banco de dados, e muito mais. O Visual C# fornece um editor de códigos avançado, designers de interface de usuário convenientes, depurador integrado, e muitas outras ferramentas para facilitar o desenvolvimento de aplicativos baseados na linguagem C# e no .NET *Framework* (INTRODUCTION, 2016).

A linguagem de programação C# oferece recursos para o desenvolvimento de programas para utilização em um computador pessoal ou notebook, aplicativos para tablets e smartphones, aplicativos UWP³², *server-side* ou *back-end*³³ de sistemas web ou sites através da integração com a plataforma de desenvolvimentos web ASP.NET e, integração com banco de dados (SQL e MySQL). Para desenvolver utilizando C#, deve-se utilizar a IDE Visual Studio, que é a ferramenta oficial da Microsoft para desenvolvimento através do *framework* .NET.

²⁹ *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado). Um programa que contém ferramentas para auxiliar no desenvolvimento de programas. Normalmente são capazes de invocar comandos do compilador para gerar e executar o programa desenvolvido.

³⁰ Módulo de extensão. Adiciona funções a outros programas.

³¹ *Framework* é uma abstração que une códigos comuns entre vários projetos de software pertencentes a um mesmo domínio de problema.

³² *Universal Windows Platform* (Plataforma Universal do Windows). Aplicativos que são adquiridos através da Windows Store.

³³ *Server-side* ou *back-end* é a linguagem de programação que o servidor web processa e envia o resultado deste processamento para o navegador, sendo exibido pela linguagem de marcação HTML, que é chamado de *client-side* ou *front-end*.

Conforme Deitel H. *et. al.* (2003, p. 60), a execução de um programa escrito na linguagem C# ocorre dentro do *framework* .NET, especificamente em sua parte central que é o CLR (*Common Language Runtime* – ambiente de tempo de execução de linguagens comum). O CLR compila o programa no momento da execução, adaptando-o ao sistema operacional.

Pensando em interoperabilidade de aplicativos, a Microsoft disponibiliza junto ao Visual Studio o programa Xamarin, que oferece a possibilidade de desenvolver aplicativos, utilizando a linguagem C#, para dispositivos móveis Android (linguagem nativa é o Java), para Windows Phone (linguagem nativa é o C#) e para iOS (linguagem nativa é o Swift). Ela também criou a .NET Foundation (<https://dotnetfoundation.org>) que, disponibiliza o código fonte da plataforma gratuitamente, além de projetos que visam melhorar a experiência de desenvolvimento de sistemas, programas e aplicativos.

A programação orientada a objetos é utilizada no desenvolvimento e implementação de programas, aplicativos e sistemas de pequeno e grande porte. As linguagens de programação orientadas a objetos são, em sua maioria, linguagens de alto nível³⁴ e contém classes que interagem entre si. As classes contém atributos e métodos que definem características e comportamento das instâncias das classes. Essas propriedades devem ser documentadas para facilitar o desenvolvimento e manutenção da aplicação desenvolvida. Essa documentação é realizada através da modelagem orientada a objetos.

2.5 MODELAGEM ORIENTADA A OBJETOS

A modelagem orientada a objetos é uma documentação gráfica que contém especificações de um projeto. O projeto pode ser um pequeno e simples aplicativo para smartphone ou computador pessoal ou, um grande e complexo sistema de gerenciamento financeiro de uma empresa multinacional que, devem ser desenvolvidos utilizando a programação orientada a objetos. Independente da utilização, tamanho e complexidade do projeto, ele deve ser modelado, com a finalidade de agilizar e garantir a integridade do processo de desenvolvimento e do processo de manutenção (correção de erros, modificações e atualizações).

Como exemplo, pode-se citar a construção de uma casa. A casa é o projeto e, para realização do projeto, são necessários documentos que contém especificações para sua

³⁴ Linguagem de alto nível possui instruções e comandos que não são interpretados diretamente pelo hardware do equipamento, necessitando de um compilador para transformar a linguagem em instrução de máquina.

construção. Um destes documentos é o memorial descritivo, um documento de texto que contém especificações sobre o material utilizado na construção das paredes, telhado, pisos, etc. Outro documento é a planta baixa, que é um desenho da casa olhando-a por cima, que contém as medidas dos cômodos, que será utilizada pelo mestre de obras, pedreiros e ajudantes para erguer as paredes da casa. Ainda existem as plantas hidráulica e elétrica, que demonstram especificamente onde passará a tubulação de água e os conduítes para passagem dos cabos elétricos. Agora, imagine a situação: certo dia o morador de uma casa resolve colocar um exaustor em sua cozinha, mas ele não consulta as plantas hidráulica e elétrica, ou pior, ele não possui estes documentos. O morador não saberá exatamente onde é possível quebrar a parede para instalação do exaustor, então ele corre o risco de interromper um conduíte, causando um curto-circuito ou interromper a tubulação hidráulica e causar uma inundação em sua cozinha. De qualquer maneira, o prejuízo será muito grande para este cidadão se ele não adivinhar o local correto para fazer o vão na parede.

Para que o desenvolvedor não faça suposições e não tente adivinhar como será o software após sua intervenção e, prejuízos não ocorram durante o desenvolvimento e manutenção deste, utiliza-se a modelagem orientada a objetos.

A modelagem é uma parte central de todas as atividades que levam à implantação de um bom software. Construimos modelos para comunicar a estrutura e o comportamento desejados do sistema. Construimos modelos para visualizar e controlar a arquitetura do sistema. Construimos modelos para compreender melhor o sistema que estamos elaborando, muitas vezes expondo oportunidades de simplificação e reaproveitamento. Construimos modelos para gerenciar os riscos (BOOCH; RUMBAUGH; JACOBSON, 2005, p. 3–4).

A criação de modelos³⁵ permite entender o software antes de seu desenvolvimento. Este entendimento é muito importante para desenvolver um software de qualidade, que seja rápido, economize recursos do equipamento onde será utilizado, seja seguro e atenda as expectativas do cliente. De posse da modelagem, o desenvolvedor poderá desenvolver um software, assim como um mestre de obras, de posse da planta baixa, pode construir uma casa. Mas como é realizada a modelagem de um software? Atualmente, ela é feita através da UML (*Unified Modeling Language*).

A UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão

³⁵ Modelo é uma simplificação da realidade. (GRADY BOOCH; JAMES RUMBAUGH e; IVAR JACOBSON, 2005, p. 6)

para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis (BOOCH; RUMBAUGH; JACOBSON, 2005, p. XIII).

A modelagem de um software, realizada através da UML, descreve através de diagramas a utilização, comportamentos e estados de um software em função de ações do usuário, do hardware, sub-rotinas ou de outros softwares. Conforme Booch; Rumbaugh; Jacobson (2005, p. XVI–XVIII), alguns anos após a chegada da programação orientada a objetos na década de 1980, surgiram métodos de análise de projetos que buscavam simplificar o entendimento, desenvolvimento, implementação e manutenção de softwares e sistemas, alguns muito complexos, desenvolvidos em programação orientada a objetos. Dentre vários métodos, três se destacavam: o método OOSE (*Object-Oriented Software Engineering - Engenharia de Software Orientada a Objeto*) de Jacobson; o método OMT (*Object Modeling Technique - Técnica de Modelagem de Objetos*) de Rumbaugh e; o método Boch, do próprio. No ano de 1994, iniciou-se a junção destes métodos e, em 1996, criou-se a versão 0.9 da UML.

O sucesso da UML 0.9 na comunidade de engenharia de software, possibilitou aos autores firmar um consórcio para aprimorar a metodologia. Este consórcio contou com grandes empresas, como: Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational, Texas Instruments e Unisys. O investimento destas empresas, possibilitou o lançamento da UML 1.0 e, em 14 de novembro de 1997, a OMG (*Object Management Group – Grupo de Gerenciamento de Objetos*), formalizou a UML 1.1 como padrão de modelagem.

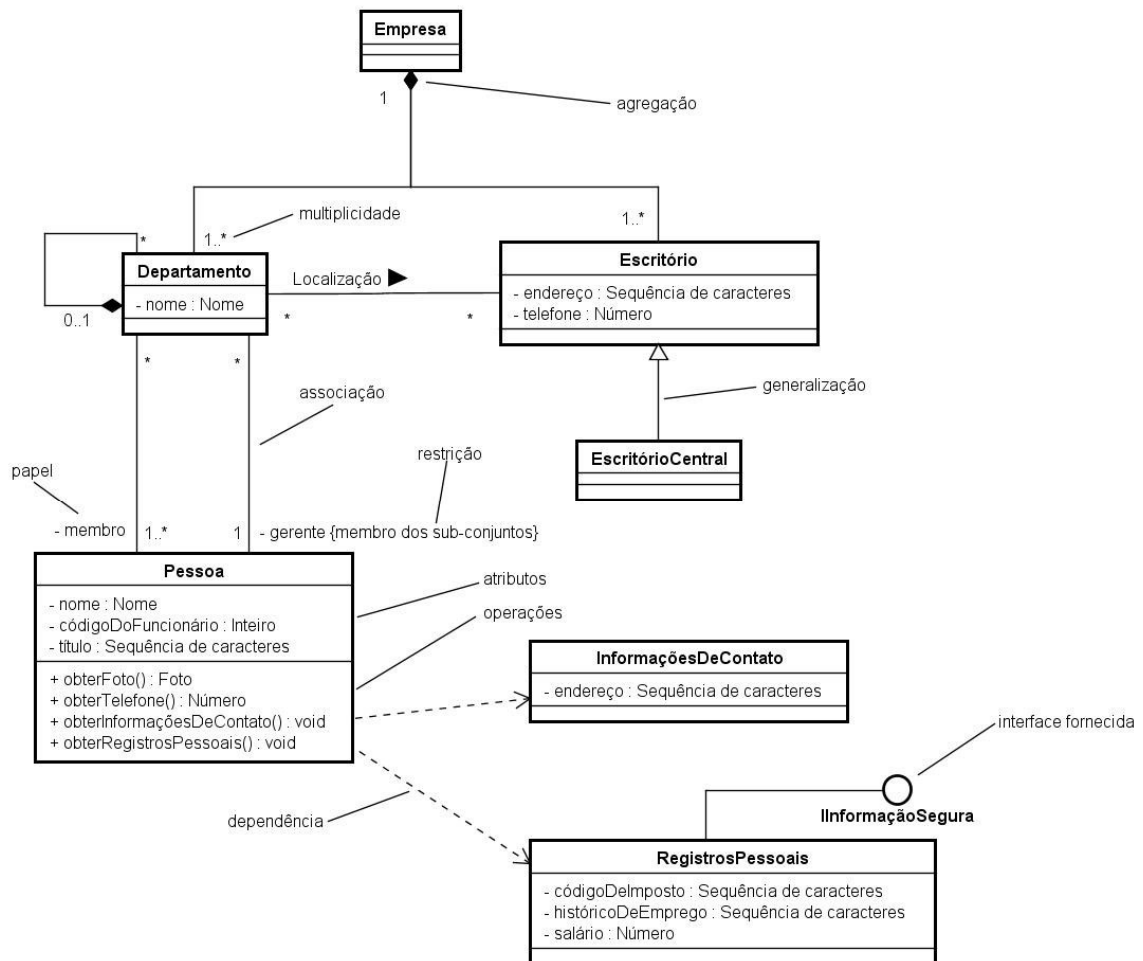
Atualmente a UML está na versão 2.0 e, de acordo com Booch; Rumbaugh; Jacobson (2005, p. 441–442), ela possui treze tipos de diagramas para modelagem, que são: diagrama de classes, diagrama de objetos, diagrama de componentes, diagrama de estrutura composta, diagrama de casos de uso, diagrama de sequências, diagrama de comunicação, diagrama de gráfico de estados, diagrama de atividades, diagrama de implantação, diagrama de pacotes, diagrama de temporização e diagrama de visão geral de interação.

Nem todos os softwares necessitam ou comportam todos estes diagramas. A seguir, serão apresentados alguns destes diagramas.

2.5.1 DIAGRAMA DE CLASSES

Um diagrama de classes exibe um modelo estático do software. Este diagrama contém classes, interfaces e seus relacionamentos. Apresentam os requisitos funcionais do sistema (*Cf.:* BOOCH; RUMBAUGH; JACOBSON, 2005, p. 108–109).

Figura 4: Exemplo de diagrama de classes

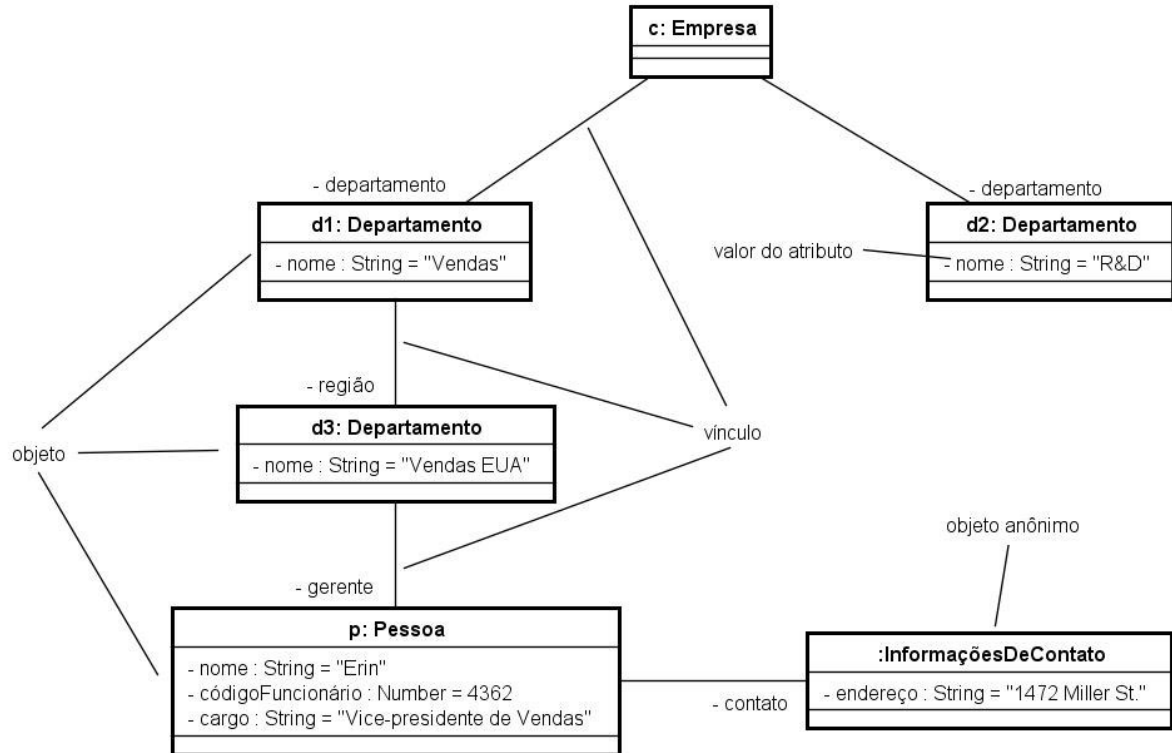


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 108)

2.5.2 DIAGRAMA DE OBJETOS

O diagrama de objetos é uma variação do diagrama de classes. Ele apresenta uma visão estática de instâncias de classes do sistema em um ponto de tempo (Cf.: BOOCH; RUMBAUGH; JACOBSON, 2005, p. 188).

Figura 5: Exemplo de diagrama de objetos



Fonte: Booch; Rumbaugh; Jacobson (2005, p. 189)

2.5.3 DIAGRAMA DE COMPONENTES

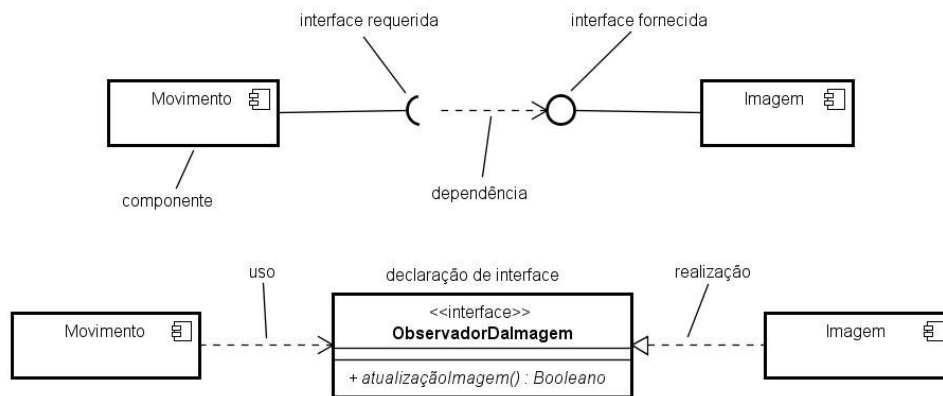
Um componente pode ser uma classe ou componentes de uma classe. Os componentes implementam interfaces. Uma interface é uma coleção de operações que definem o comportamento de um componente. O relacionamento entre componente e interface permite que um componente possa ser substituído em tempo de execução. Um componente pode ser reutilizado em outros sistemas (Cf.: BOOCH; RUMBAUGH; JACOBSON, 2005, p. 195–199).

De acordo com Booch; Rumbaugh; Jacobson (2005, p. 206), um exemplo da utilização de componentes e interfaces é uma API (*Application Program Interface* – Interface de Programação de Aplicações) e, como exemplo de API, pode-se expor um sistema de pagamento on-line por cartão de crédito. Basta fazer o download da API no site da financeira e conectá-la ao sistema através de um método fornecido, definindo-se os atributos como nome do beneficiário, número de conta, etc. Ou ainda de maneira simplificada, apenas fornecendo

ao método a chave de conexão (uma sequência de números e letras) fornecida pela financeira, para que a API acesse as informações do beneficiário no banco de dados desta.

A figura 7 apresenta duas representações de relacionamento. A primeira com a interface oculta e a segunda com a interface expandida.

Figura 6: Exemplo de diagrama de componentes com interfaces

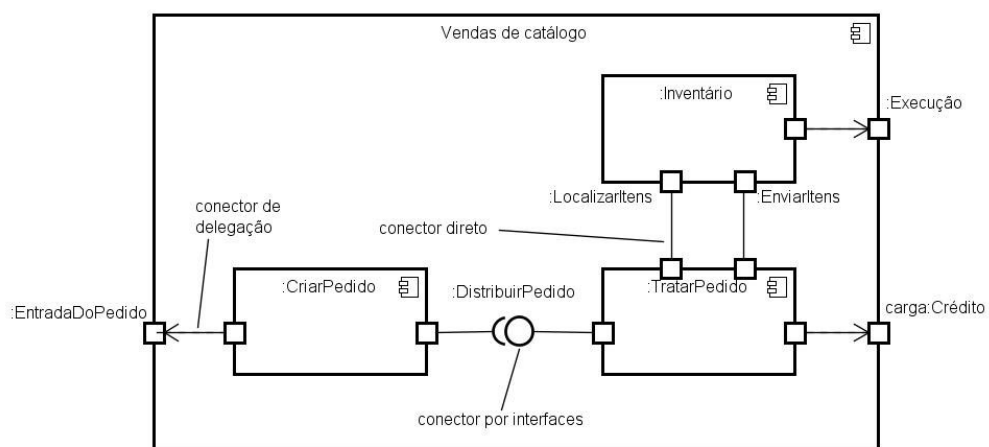


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 198)

2.5.4 DIAGRAMA DE ESTRUTURA COMPOSTA

Os autores da UML, tratam o diagrama de estrutura composta junto com o diagrama de componentes, visto que ambos mostram conectores, portas e a estrutura interna das classes que formam um componente. A diferença entre os diagramas é muito sutil (Cf.: BOOCH; RUMBAUGH; JACOBSON, 2005, p. 97).

Figura 7: Exemplo de diagrama de estrutura composta com conectores

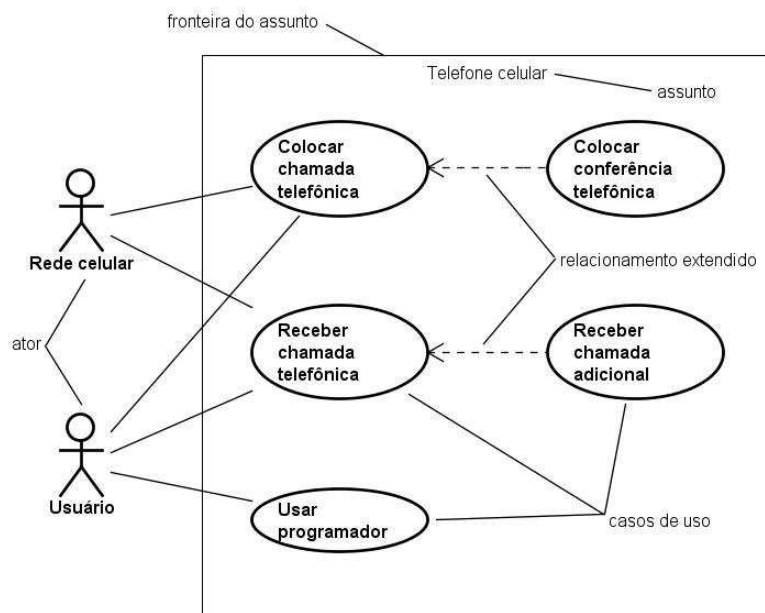


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 204)

2.5.5 DIAGRAMA DE CASOS DE USO

Um diagrama de casos de uso mostra a parte externa de um sistema, ou seja, como ele se comporta perante ações de atores (usuários ou outros sistemas). Ele é importante por facilitar o entendimento do comportamento de um elemento do sistema (Cf.: BOOCH, RUMBAUGH; JACOBSON, 2005, p. 240).

Figura 8: Exemplo de diagrama de casos de uso



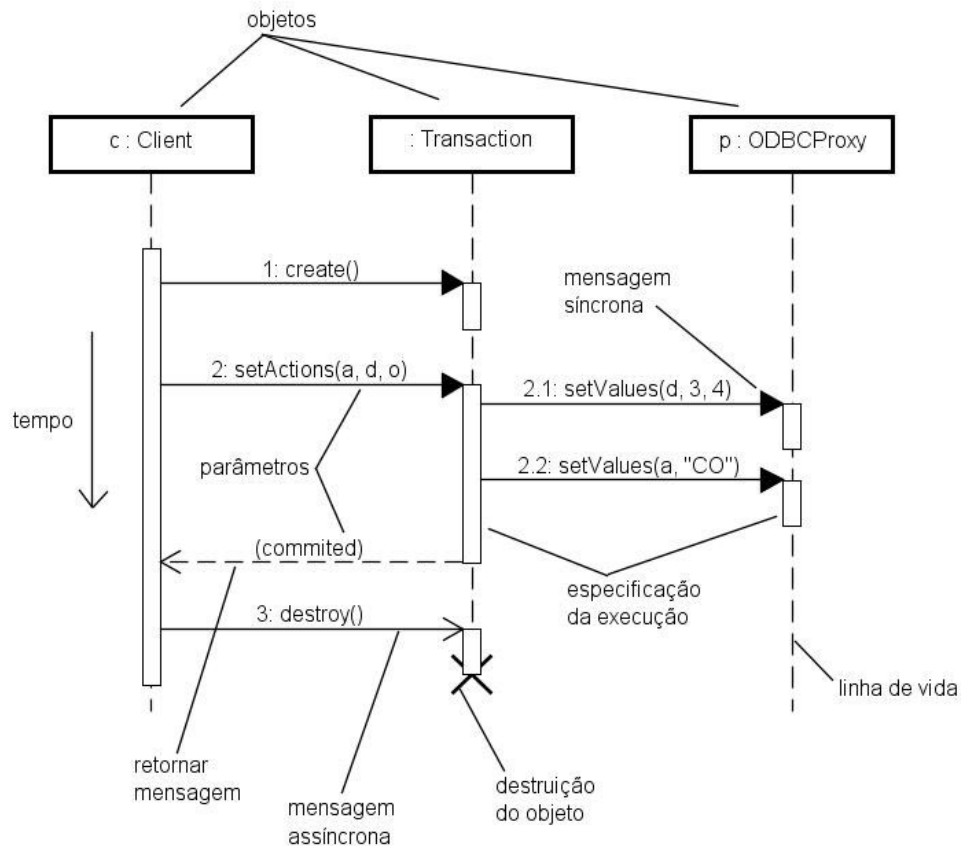
Fonte: Booch; Rumbaugh; Jacobson (2005, p. 242)

2.5.6 DIAGRAMA DE SEQUÊNCIAS

Um diagrama de sequência exibe os objetos, as trocas de informações entre eles e o seu tempo de vida dentro da interação. Conforme Booch; Rumbaugh; Jacobson (2005, p. 99):

Um diagrama de sequência é um diagrama de interação que dá ênfase à ordenação temporal de mensagens. Um diagrama de sequência mostra um conjunto de papéis e as mensagens enviadas e recebidas pelas instâncias que representam os papéis. Use os diagramas de sequência para ilustrar a visão dinâmica de um sistema.

Figura 9: Exemplo de diagrama de seqüências



Fonte: Booch; Rumbaugh; Jacobson (2005, p. 255)

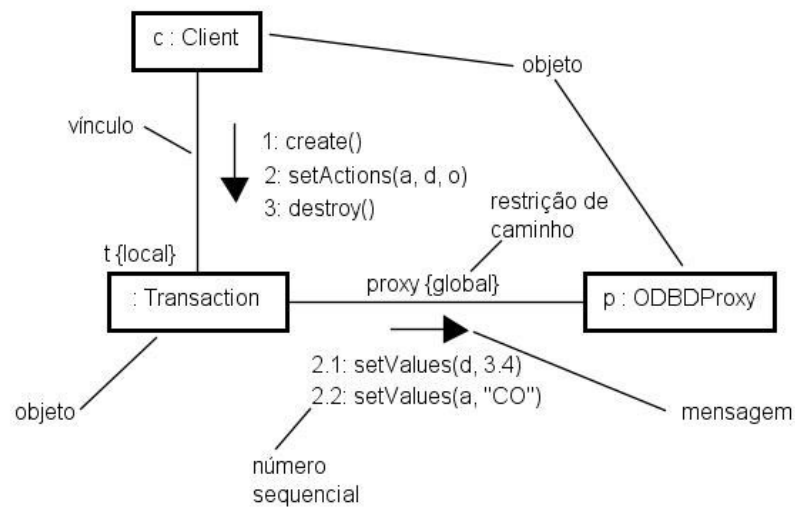
2.5.7 DIAGRAMA DE COMUNICAÇÃO

De acordo com Booch; Rumbaugh; Jacobson (2005, p. 99):

Um diagrama de comunicação é um diagrama de interação que dá ênfase à organização estrutural dos objetos que enviam e recebem mensagens. Um diagrama de comunicação mostra um conjunto de papéis, as conexões existentes entre esses papéis e as mensagens enviadas e recebidas [pelas] instâncias que representam os papéis.

Podemos perceber que um diagrama de comunicação é muito semelhante ao diagrama de seqüências, exibindo a dinâmica do sistema, com a diferença de não exibir o tempo de vida da interação.

Figura 10: Exemplo de diagrama de comunicação

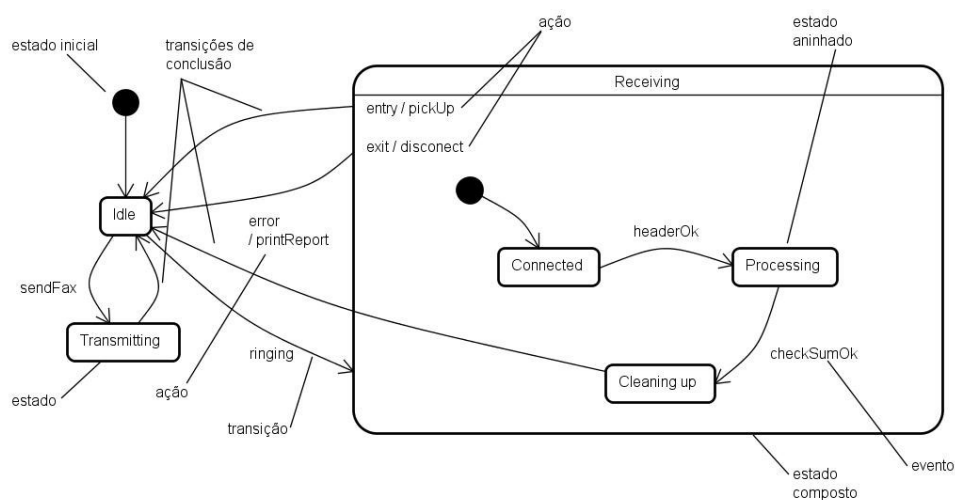


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 259)

2.5.8 DIAGRAMA DE GRÁFICO DE ESTADOS

Conforme Booch; Rumbaugh; Jacobson (2005, p. 99 e 341), um diagrama de gráfico de estados exibe uma máquina de estados, que é o tempo de vida de estados, transições, eventos e atividades. Uma máquina de estado é o comportamento de um objeto, solicitado por eventos do mesmo ou de outro objeto. Este tipo de diagrama enfatiza o fluxo de controle de um estado para outro.

Figura 11: Exemplo de diagrama de gráfico de estados

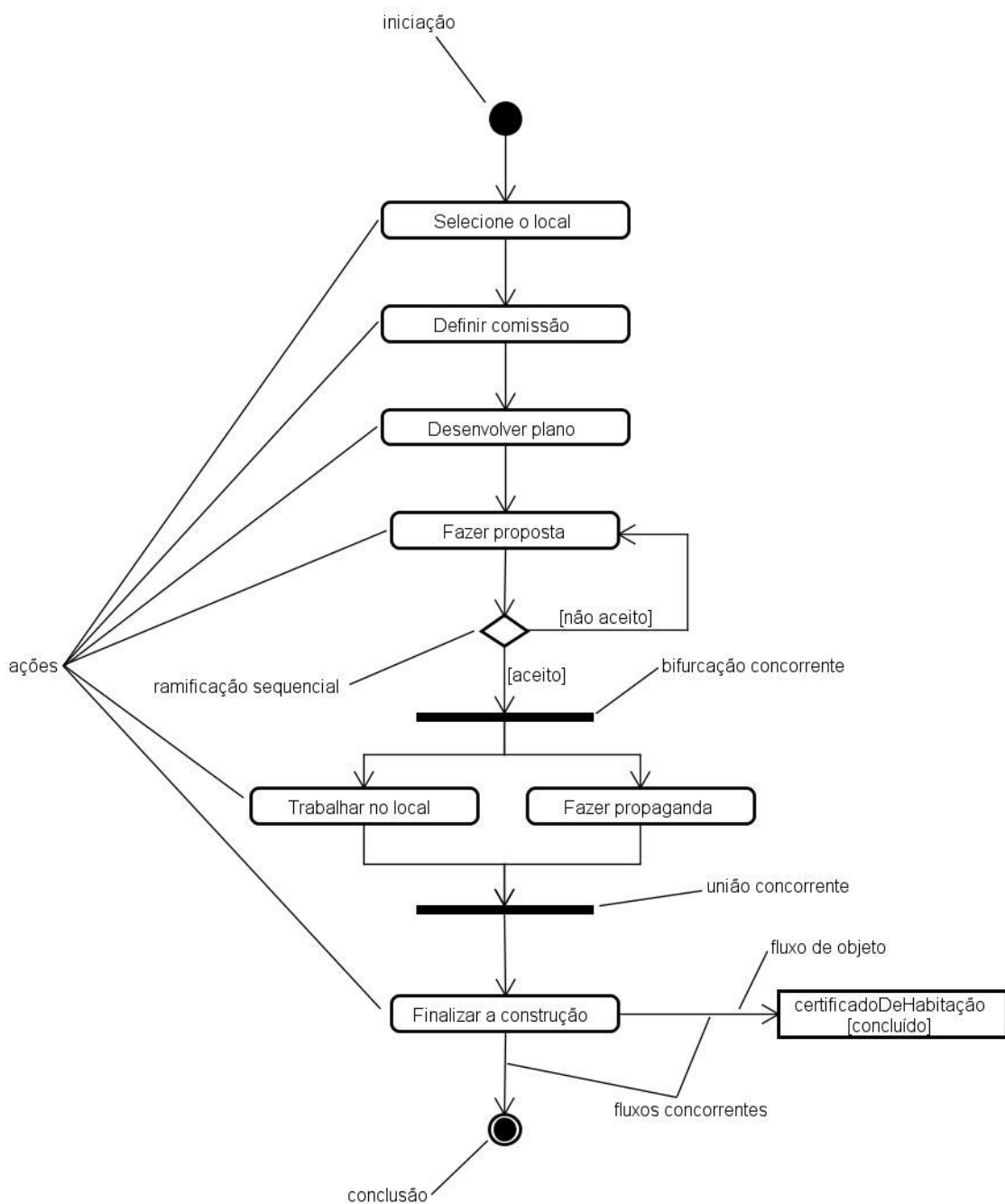


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 343)

2.5.9 DIAGRAMA DE ATIVIDADES

Um diagrama de atividades apresenta o fluxo de controle entre atividades ou o fluxo de estados de um objeto. Este fluxo pode ser sequencial, uma atividade após a outra, ou concorrente, com atividades ocorrendo ao mesmo tempo (Cf.: BOOCH, RUMBAUGH; JACOBSON, 2005, p. 268).

Figura 12: Exemplo de diagrama de atividades

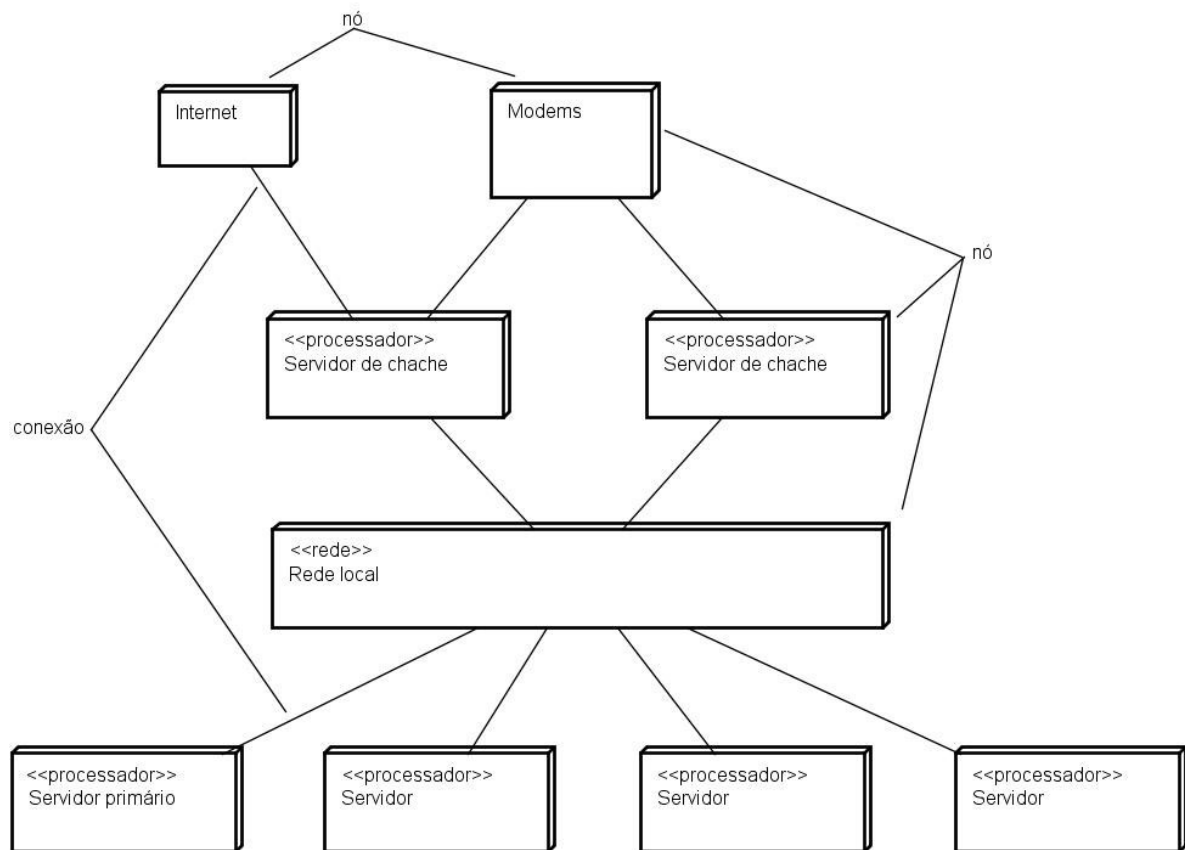


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 270)

2.5.10 DIAGRAMA DE IMPLANTAÇÃO

Um diagrama de implantação é utilizado para a modelagem física do sistema, seja apenas um hardware ou o local de sua localização (Cf.: BOOCH, RUMBAUGH; JACOBSON, 2005, p. 411).

Figura 13: Exemplo de diagrama de implantação

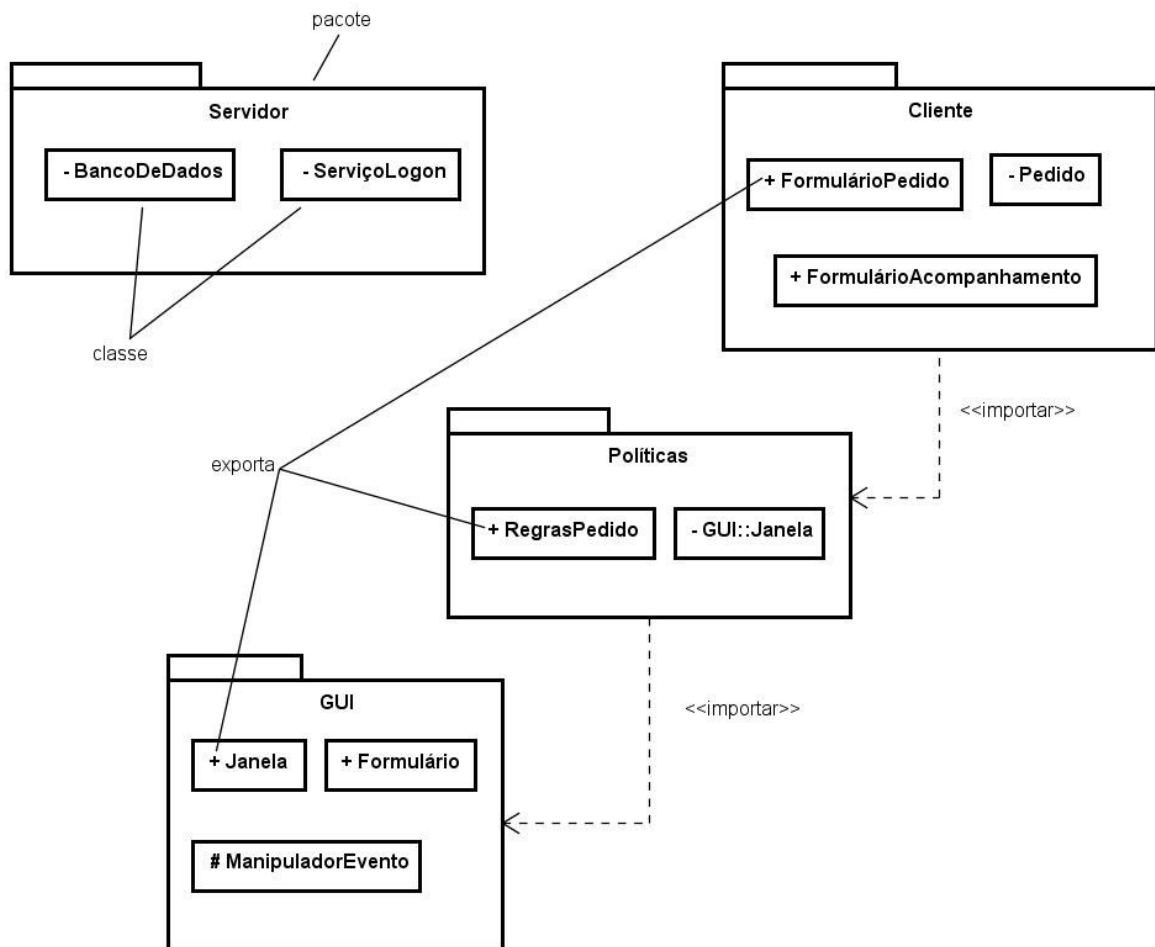


Fonte: Booch; Rumbaugh; Jacobson (2005, p. 412)

2.5.11 DIAGRAMA DE PACOTES

Um diagrama de pacotes representa a organização do sistema em agrupamentos lógicos. Pode-se, por exemplo, organizar as classes, interfaces, componentes e diagramas em pacotes separados (Cf.: BOOCH, RUMBAUGH; JACOBSON, 2005, p. 167).

Figura 14: Exemplo de diagrama de pacotes



Fonte: Booch; Rumbaugh; Jacobson (2005, p. 173)

A modelagem de um sistema é de extrema importância para auxiliar desenvolvedores, analistas e auditores no entendimento e execução de tarefas relacionadas a aplicação. Uma aplicação pode ser um pequeno programa, um aplicativo para dispositivo móvel ou uma grande aplicação financeira. Quanto maior a complexidade da aplicação, maiores serão os problemas causados pela falta de documentação e modelagem.

Aplicações Web também são modeladas e, são utilizadas cada vez mais em residências, instituições de ensino e instituições corporativas. Visto a importância da Web para o desenvolvimento de aplicações, faz-se necessário um estudo desta.

2.6 WEB

Acessar um site para informar-se acerca de política, saúde e educação, consultar quais filmes estão em cartaz no cinema, acessar o site da faculdade para fazer o download do conteúdo de aula compartilhado pelo professor, acessar uma rede social para conversar e compartilhar conteúdo com familiares e amigos, comprar um caderno, um computador ou qualquer outra mercadoria que possa ser entregue em seu endereço, tudo isto é possível porque existe a internet e a Web.

Conforme W3C (2017), a internet é a uma rede de computadores conectados que transmitem dados através da comutação de pacotes e utilizam o conjunto de protocolos padronizados TCP/IP (*Transmission Control Protocol/Internet Protocol* – Protocolo de Controle de Transmissão/Protocolo de Internet). Web é uma referência a WWW (*World Wide Web* – Rede Mundial de Computadores), que é um espaço de informação que possui um identificador denominado URI (*Uniform Resources Identifiers* – Identificador Uniforme de Recursos). Para acessar um site, utiliza-se um tipo de URI denominado URL (*Uniform Resource Locator* – Localizador Padrão de Recursos), que deve ser digitado na barra de endereços de um navegador para que um protocolo específico, o HTTP (*Hyper Text Transfer Protocol* – Protocolo de Transferência de Hipertexto), envie para a Web a requisição de acesso ao site e, o servidor onde o site está hospedado seja localizado e devolva os recursos ou informações requisitadas, o site para exibição no navegador. Um exemplo de URL: <http://www.fatec.edu.br>. Ainda existem outros tipos de protocolos utilizados pelo URL como: o HTTPS (*Hyper Transfer Protocol Secure* – Protocolo de Transferência de Hipertexto Seguro), que combina os protocolos HTTP e SSL (*Secure Socket Layer* – Camada de Ligação Segura) para criptografar os dados de comunicação entre um site e o navegador, sendo utilizado principalmente para operações bancárias e compras pela web e; o protocolo FTP (*File Transfer Protocol* – Protocolo de Transferência de Arquivos, utilizado para transferir arquivos entre cliente e servidor, necessitando de autenticação com login e senha, podendo ser realizada por um navegador ou por um programa específico para esta operação.

A Web foi inventada em 1989 por Tim Berners-Lee, físico britânico formado pela universidade de Oxford que trabalhava no CERN (*Conseil Européen pour la Recherche Nucléaire* – Organização Europeia para a Pesquisa Nuclear). O projeto inicial da Web, que se chamava “*Mesh*”, era o compartilhamento em tempo real de informações de projetos entre cientistas de institutos e universidades de qualquer parte do planeta. O primeiro servidor foi

um computador NeXT³⁶, pertencente a Tim Berners-Lee, que escreveu e disponibilizou o primeiro site, contendo informações do projeto. O CERN disponibiliza o site original através do URL: <http://info.cern.ch/hypertext/WWW/TheProject.html> (Cf.: CERN, 2017). Hoje, Tim Berners-Lee é diretor da W3C (*World Wide Web Consortium* – Consórcio da Rede Mundial de Computadores), uma organização que gerencia os padrões da Web e desenvolve tecnologias para ampliá-la e aperfeiçoá-la (Cf.: W3C, 2017).

2.6.1 Site

Um site, ou website, é um conjunto de páginas Web, que são escritas através de textos organizados por uma marcação, o HTML (*Hypertext Markup Language* – Linguagem de Marcação de Hipertexto). W3C (2017) diz que “HTML é a linguagem para descrever a estrutura de páginas Web”³⁷.

A linguagem HTML formata textos como parágrafos, cabeçalhos, citações, *hiperlinks*, exibe imagens, vídeos, desenhos vetoriais, mapas e cria listas, formulários e tabelas, tudo isto através de *tags* específicas para cada item citado. O exemplo abaixo mostra a marcação de um parágrafo através da *tag* de abertura `<p>` e da *tag* de fechamento `</p>`. Somente o texto que está entre estas *tags* é exibido no navegador.

```
<p>Isto é um parágrafo.</p>
```

Uma *tag* pode conter tipos de atributos, por exemplo, atributos de identificação para estilização (*class* - classe) e atributos de identificações para implementações em linguagens de *script* ou programação (*id*). Ambos atributos podem ser utilizados para estilização ou programação, mas o uso indicado foi descrito anteriormente. Veja um exemplo de *tag* com atributo de estilização:

```
<p class="referencia">Isto é um parágrafo com atributo de
estilização.</p>
```

Este atributo é importante para que a folha de estilos possa encontrar a *tag* e personaliza-la, por exemplo, alterando a cor, tamanho e posição do texto. A folha de estilos é um arquivo que fica separado e referenciado no arquivo HTML e é chamado de CSS

³⁶ Os computadores NeXT foram criados por Steve Jobs em 1985, quando deixou a Apple Inc.. Destes computadores, nasceu o sistema operacional MacOS X da Apple Inc., que na NeXT, era chamado de Mach.

³⁷ HTML is the language for describing the structure of Web pages. (W3C, 2017. *Tradução nossa*)

(*Cascading Style Sheets* – Folhas de Estilo em Cascata). Veja abaixo, o exemplo da estilização do parágrafo anterior:

```
.referencia {  
    color: #ff0000;  
    font-size: 20px;  
}
```

A classe *referencia* escrita no arquivo HTML é invocada no arquivo CSS escrevendo o símbolo (.) ponto e o nome da classe, seguido de chaves de abertura e fechamento. Entre as chaves, fica o bloco de estilização que é composto por atributos do elemento referenciado (*color* e *font-size*) e pelas propriedades dos atributos (*#ff0000* e *20px*). Este bloco de estilização pode conter um ou muitos atributos e, estes atributos, podem ser comuns ou específicos aos elementos HTML.

CSS é o idioma para descrever a apresentação de páginas da Web, incluindo cores, layout e fontes. Ele permite adaptar a apresentação a diferentes tipos de dispositivos, como telas grandes, telas pequenas ou impressoras. O CSS é independente do HTML e pode ser usado com qualquer linguagem de marcação baseada em XML. A separação entre HTML e CSS facilita a manutenção de sites, o compartilhamento de folhas de estilo entre páginas e a personalização de páginas em diferentes ambientes³⁸ (W3C, 2017).

A personalização da página Web pela folha de estilos, vai além da troca de cores e tamanho do texto. Ela permite que o desenvolvedor crie regras de estilização para cada tipo de dispositivo e resolução de tela, fazendo com que o conteúdo seja exibido corretamente e de forma distinta em dispositivos diferentes, como um laptop, um *smartphone* ou um computador pessoal. Permite também que se realize animações dos elementos HTML como, mudança de cores, alteração do tamanho e posição. Estas animações podem ocorrer através da interação do usuário da página Web, um clique ou posicionamento do cursor do mouse, ou podem ser carregadas automaticamente na abertura da página ou após um tempo determinado pelo desenvolvedor.

Algumas das informações presentes em um site, necessitam de um armazenamento gerenciado, que permita uma interação com o usuário. Este tipo de informação fica armazenada em um banco de dados, por isso a seção seguinte estuda o sistema de gerenciamento de banco de dados MySQL.

³⁸ CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments (W3C, 2017. *Tradução nossa*)

2.7 MYSQL

MySQL é um sistema de gerenciamento de banco de dados (SGBD) ou *database management system* (DBMS), desenvolvido, distribuído e apoiado pela Oracle Corporation. Este sistema gerencia bancos de dados de linguagem de consulta estruturada (SQL - *Structured Query Language*), uma linguagem padronizada desenvolvida em 1986.

Um banco de dados é uma coleção estruturada de dados. Um dado pode ser qualquer coisa, uma simples lista de compras, uma galeria de imagens ou as enormes quantidades de informações em uma rede corporativa. Para adicionar, acessar e processar dados armazenados em um banco de dados de computador, você precisa de um sistema de gerenciamento de banco de dados, como o *MySQL Server*. Como os computadores são muito bons em lidar com grandes quantidades de dados, os sistemas de gerenciamento de banco de dados desempenham um papel central na computação, como utilitários autônomos ou como partes de outras aplicações³⁹ (ORACLE, 2017).

Um banco de dados relacional armazena valores de diferentes tipos, como: texto (nome, cidade, profissão, etc), número (identificação, valor monetário, nota de prova, etc), tempo (data e hora), binário (imagem, vídeo e som) e espacial (dados geométricos e dados geográficos). O gerenciamento destes dados, realizado pelo MySQL ou por outro SGBD⁴⁰, permite que aplicações desenvolvidas em linguagens de programação como o C#, Java e PHP acessem grandes volumes de informação com rapidez, integridade e consistência.

Conforme Oracle (2017), um banco de dados é um arquivo físico que busca a otimização de velocidade e utiliza tabelas para organizar os dados em campos personalizáveis que podem se relacionar uns com os outros e com campos de outras tabelas. As tabelas permitem que o banco de dados seja organizado de acordo com a atividade e agilizam a consulta aos dados, já que a requisição é feita em uma tabela ou, em um conjunto de tabelas, e não no banco de dados completo. Por exemplo, em uma escola, pode-se montar um banco de dados com as tabelas: professor e curso. A tabela professor armazena os dados pessoais dos professores da escola (número de identificação, nome, data de nascimento e e-mail) e a tabela cursos, armazena os cursos oferecidos pela escola (número de identificação, nome, descrição e duração). A exibição de uma lista de cursos oferecidos pela escola é feita realizando-se uma

³⁹ A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications. (ORACLE CORPORATION, 2017. *Tradução nossa*)

⁴⁰ SQL Server, Oracle, PostgreSQL e MariaDB Server são exemplos de SGBD.

consulta a tabela cursos, a tabela professores não precisa ser consultada. Agora, para saber quais professores lecionam um curso específico, é necessário fazer um relacionamento entre as tabelas. Para relacionar um professor com um curso, uma maneira é criando-se uma tabela de relacionamento, que contém apenas dois campos: o número de identificação do professor e o número de identificação do curso. Este relacionamento permite que um professor possa ter muitos cursos e um curso possa ser lecionado por muitos professores. Além disso, uma alteração no nome do curso, por exemplo, se propaga a todos os professores que possuam relação com este curso, eliminando a inconsistência de dados, já que a alteração ocorre em apenas um campo de uma única tabela. A rapidez de acesso, a consistência e integridade de dados dependem do desenvolvedor que deve seguir boas práticas de desenvolvimento, e dependem principalmente de um bom SGBD.

Oracle (2017) ressalta que o MySQL consegue lidar com grandes bancos de dados com velocidade e segurança, porque foi desenvolvido com este propósito e também oferece um rico conjunto de funções para garantir estes requisitos. MySQL funciona em sistemas cliente/servidor e também em sistemas embarcados.

O MySQL é de grande importância para o gerenciamento de um banco de dados e muitas linguagens de programação oferecem métodos de conexão com este sistema. O *framework* .NET dispõe de recursos que facilitam a integração com o MySQL e utilização com as linguagens de programação aceitas pela plataforma. A seção seguinte apresenta o *framework* .NET e sua importância para o desenvolvimento de aplicações.

2.8 .NET

O *framework* .NET (lê-se *dot net*) é um ambiente de execução gerenciado, lançado no ano de 2002 pela Microsoft que, inicialmente, tinha como objetivo a implementação de aplicações web e programas para o sistema operacional Windows. Atualmente, com o .NET é possível desenvolver aplicações para computadores, *smartphones*, *tablets*, web e grandes sistemas computacionais e, estas aplicações podem ser utilizadas em qualquer sistema operacional que possua o ambiente de execução .NET instalado.

Com o .NET é possível utilizar diversas linguagens de programação para desenvolver aplicações, entre elas: C#, C++, Visual BASIC .NET, JScript (o JavaScript da Microsoft) e J# (uma versão de Java da Microsoft) (Cf.: SEBESTA, 2010, p. 52). Então, como o .NET consegue executar aplicações escritas em linguagens de programação diferentes em sistemas operacionais distintos?

O *framework* .NET é um ambiente de execução gerenciado que provê uma variedade de serviços para a execução de aplicações. Ele consiste em dois componentes principais: a linguagem comum em tempo de execução (CLR), que é o motor que gerencia a execução de aplicativos e a biblioteca de classes do .NET, que provê uma biblioteca de código testado e reutilizável, que os desenvolvedores podem implementar em suas aplicações⁴¹ (GETTING. 2017).

Conforme Deitel H. *et. al.* (2003, p. 60), o .NET compila o programa escrito em uma linguagem específica para uma linguagem intermediária da Microsoft, o MSIL ou IL (*Microsoft Intermediate Language*). Em seguida, o CLR realiza a compilação JIT (*just-in-time*), ou seja, em tempo de execução, para código de máquina. Como esta compilação é realizada em tempo de execução, basta que o sistema operacional possua o .NET instalado para que o programa possa ser executado, sendo assim, é possível executar o programa em diferentes sistemas operacionais.

Conforme .NET (2017), o .NET está em sua versão 4.7 e, a partir da versão 3.0, passou a ser distribuído juntamente com o sistema operacional que, na época, era o Windows Vista. Em sistemas operacionais anteriores, desatualizados ou diferentes do Windows, deve-se fazer o download gratuito do .NET que, é distribuído em arquivos com as extensões: exe e zip para Windows; pkg para macOS e; tar.gz para Linux.

Para desenvolver aplicações .NET, a Microsoft disponibiliza o IDE Visual Studio, que oferece suporte para desenvolvimento, depuração e compilação de aplicações e, ele também possui suporte a diversas extensões, entre elas o Xamarin, com a qual é possível desenvolver aplicativos multiplataformas para dispositivos móveis Android, iOS e Windows Phone, utilizando a linguagem de programação C#.

Existe também o Mono, uma implementação de código aberto do .NET patrocinada pela Microsoft, que permite o desenvolvimento de aplicações em sistemas operacionais Linux e macOS através do IDE MonoDevelop, que também pode ser utilizado no sistema operacional Windows.

⁴¹ The .NET Framework is a managed execution environment that provides a variety of services to its running applications. It consists of two major components: the common language runtime (CLR), which is the execution engine that handles running applications; and the .NET Framework Class Library, which provides a library of tested, reusable code that developers can call from their own applications. (GETTING STARTED WITH THE .NET FRAMEWORK. 2017. *Tradução nossa*)

2.8.1 ASP.NET

O desenvolvimento de aplicações web na plataforma .NET é feito através do ambiente ASP.NET. Este ambiente é uma atualização do ambiente ASP (*Active Server Page* – Página do Servidor Ativo).

ASP.NET é uma plataforma Web que fornece todos os serviços que você precisa para construir aplicações web profissionais baseadas em servidor. ASP.NET é baseado em .NET *Framework*, portanto, todos os recursos .NET *Framework* estão disponíveis para as aplicações ASP.NET. Os aplicativos podem ser escritos em qualquer linguagem compatível com o *Common Language Runtime* (CLR), incluindo Visual Basic e C# (ASP.NET, 2017).

Com o ASP.NET, cria-se um arquivo que é exibido em um navegador web. Este arquivo possui marcações HTML (*client-side*), que são interpretadas pelo navegador web e componentes ASP.NET, que são implementados por uma linguagem de programação (*server-side*) compatível com o CLR e interpretados pelo servidor. A interpretação realizada no servidor devolve ao navegador uma marcação HTML, por exemplo: um componente ASP.NET, escrito como `<asp:TextBox>`, que recebe textos digitados pelo usuário, é exibido no código HTML da página como `<input type="text"/>`, ou seja, uma marcação HTML (Cf.: GLOSSÁRIO, 2017).

A linguagem de programação utilizada no desenvolvimento da aplicação, pode ser implementada no mesmo arquivo que contém as marcações HTML ou em um arquivo separado, próprio da linguagem utilizada. Esta linguagem de programação pode realizar implementações dinâmicas para exibições de elementos no navegador, cálculos matemáticos ou requisições em um banco de dados.

Conforme ASP.NET (2014), existem três tecnologias para criação de aplicações ASP.NET web dinâmicas: *Web Pages* (páginas web), *Web Forms* (Formulários Web) e MVC (*Model – View – Controller* - Modelo, Vista e Controlador).

Tabela 1: Tecnologias ASP.NET web dinâmicas

TECNOLOGIA	DESCRIÇÃO
<i>Web Pages</i>	Tem como ponto central a adição de código dinâmico no servidor, acesso a banco de dados e sintaxe leve e simples.
<i>Web Forms</i>	Baseia-se em um modelo de objeto de página e controles típicos de janelas <i>Windows Forms</i> , como: botões e listas. Quem desenvolve em <i>Windows Forms</i> para aplicações desktop, perceberá que <i>Web Forms</i> utiliza o mesmo modelo baseado em eventos.

MVC	Implementa o padrão <i>Model – View – Controller</i> para ASP.NET. A ênfase é a separação de interesses: processamento, dados e interface de usuário.
-----	---

Fonte: Microsoft (2014)

O desenvolvimento de aplicações implementadas pelo ambiente de desenvolvimento web ASP.NET, auxilia o desenvolvedor a criar aplicações dinâmicas e seguras, com a utilização de tecnologia e linguagem de programação à sua escolha, dentre as variedades suportadas pelo *framework* .NET. O próximo capítulo, exhibe a modelagem e implementação de um sistema web que utiliza o ambiente ASP.NET.

3 SISTEMA BLEAN

O sistema Blean é um ambiente virtual de aprendizagem (AVA) desenvolvido com o propósito de auxiliar professores na distribuição de conteúdo de aula para seus alunos. O aluno poderá utilizar esse conteúdo em sala de aula ou fora dos limites da instituição de ensino, através de um *smartphone*, *notebook* ou computador pessoal com acesso a internet. Este modelo de aprendizado, onde as aulas podem ser acessadas tanto no ambiente escolar quanto fora dele, é chamado de aprendizado misto e, por isso, o sistema foi batizado como Blean, já que aprendizado misto pode ser referenciado como *b-learning* (*blended learning*).

O sistema será on-line, sendo hospedado em um servidor web IIS (*Internet Information Services* – Serviços de Informação da Internet) de criação da Microsoft e, será desenvolvido na plataforma web ASP.NET, linguagem de programação C# e sistema de gerenciamento de banco de dados MySQL.

Qualquer pessoa que acessar o sistema, poderá cadastrar-se com informações básicas de contato e definições de *login* e senha. Este cadastro é gratuito e permite a pessoa acessar o sistema com o perfil de aluno, tendo acesso as aulas disponibilizadas por um professor. Para alteração do perfil para professor, é necessário fazer um requerimento ao administrador do sistema que, após análise, poderá fazer a transição de perfil. O professor poderá criar cursos e conteúdo de aula sem limitação de quantidade. Apenas para os casos de upload, será estabelecido um limite de 5Mb (cinco megabytes) para o tamanho individual dos arquivos. O professor deverá definir um código de inscrição para cada curso e fornecê-lo aos alunos, permitindo que eles acessem o curso. O professor poderá acessar e alterar as informações da conta de seus alunos e bloquear o acesso aos cursos. O administrador terá acesso a todos os dados dos usuários e, poderá alterar dados do usuário, excluir contas, alterar o perfil ou bloquear o acesso ao sistema.

3.1 ESPECIFICAÇÃO DOS REQUISITOS

Esta seção apresenta os requisitos funcionais e requisitos não funcionais que deverão ser implementados no ambiente virtual de aprendizagem Blean.

3.1.1 REQUISITOS FUNCIONAIS

Esta seção apresenta os requisitos essenciais para funcionamento pleno do ambiente virtual de aprendizagem Blean.

3.1.1.1 VISITANTE

1. Criar conta de usuário.
 - a. Nome completo (obrigatório).
 - b. E-mail (obrigatório).
 - c. Nome de usuário (obrigatório).
 - d. Senha (obrigatório).
 - e. Confirmação de senha (obrigatório).
2. Contatar o administrador.

3.1.1.2 ADMINISTRADOR, PROFESSOR E ALUNO

1. Alteração de informações pessoais
 - a. Confirmar senha atual (obrigatório).
2. Excluir conta.
 - a. Confirmar senha atual (obrigatório).

3.1.1.3 ADMINISTRADOR

1. Contas de usuários.
 - a. Alterar informações do usuário.
 - b. Alterar o perfil do usuário (aluno ou professor).
 - c. Bloquear/desbloquear a conta do usuário.
 - d. Apagar a conta do usuário.
2. Turmas.
 - a. Criar.
 - b. Alterar.
 - c. Excluir.
3. Cursos.
 - a. Criar.
 - b. Alterar.
 - c. Excluir.
4. Aulas.
 - a. Criar.
 - b. Alterar.
 - c. Excluir.

3.1.1.4 PROFESSOR

1. Turmas.
 - a. Criar.
 - b. Alterar.
 - c. Excluir.
2. Cursos.
 - a. Criar.
 - b. Alterar.
 - c. Excluir.
3. Aulas.
 - a. Criar.
 - b. Alterar.
 - c. Excluir.
4. Contas de alunos.
 - a. Visualizar.
 - b. Alterar.
 - c. Bloquear/desbloquear acesso ao curso.
5. Contatar o administrador.

3.1.1.5 ALUNO

1. Acessar cursos.
 - a. Primeiro acesso mediante confirmação do código de inscrição fornecido pelo professor.
2. Contatar o administrador.

3.1.2 REQUISITOS NÃO FUNCIONAIS

Esta seção apresenta os requisitos que não são essenciais para o funcionamento pleno do ambiente virtual de aprendizagem Blean.

1. A senha deverá conter entre 8 e 20 caracteres.
2. A senha poderá conter números, letras minúsculas, letras maiúsculas e caracteres especiais.
3. Senhas não ficam visíveis nas telas de informações de contato. O campo fica vazio, aguardando o usuário digitá-la para confirmação.

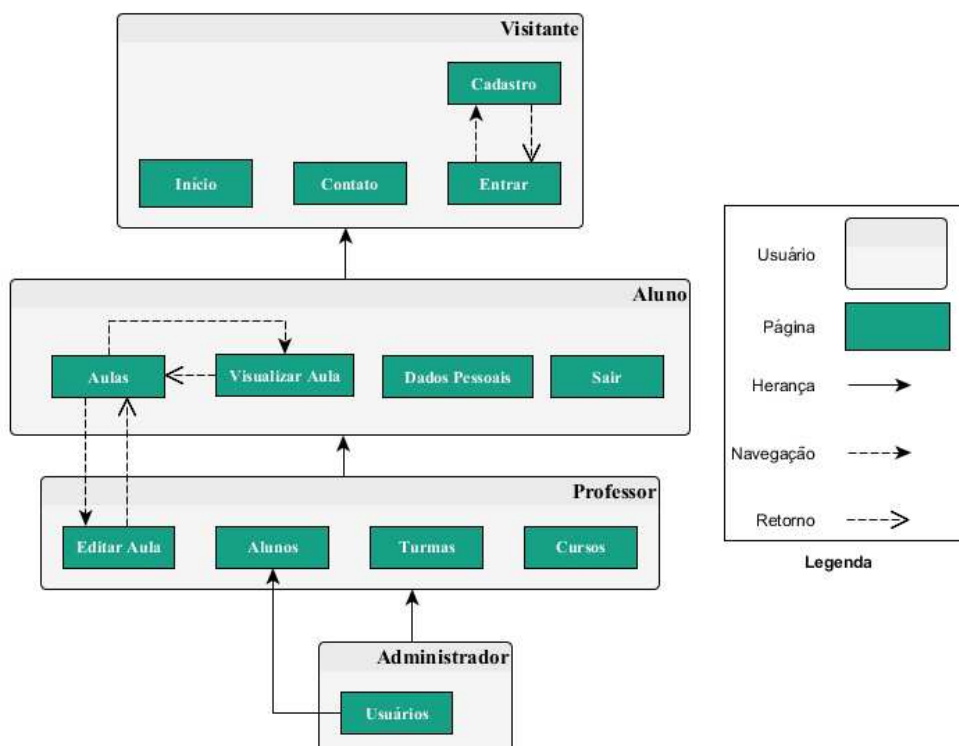
3.2 MODELAGEM

Esta seção apresenta o diagrama de navegação, os diagramas UML e os diagramas de modelagem do banco de dados relacional do sistema Blean. Para o diagrama de navegação, utilizou-se o software yEd Graph Editor, para a UML o software Astah Community e, para o banco de dados, o software brModelo.

3.2.1 DIAGRAMA DE NAVEGAÇÃO

O diagrama de navegação apresenta a estrutura de páginas e suas relações com os usuários. O usuário **Visitante** é qualquer pessoa que acesse o sistema Blean sem registro no sistema e, terá acesso a três páginas: Início, Contato e Entrar. O usuário **Aluno** é um usuário registrado que tem acesso, além das páginas do usuário Visitante, as páginas: Aulas, Visualizar Aula, Dados Pessoais e Sair (um link sair do sistema). O usuário **Professor** é um usuário registrado que tem acesso as páginas do usuário Aluno e as páginas: Editar Aula, Alunos, Turmas e Cursos. O usuário **Administrador** é um usuário registrado que tem acesso as páginas do usuário Professor com uma diferença: a página Alunos transforma-se em Usuários, para que o Administrador tenha acesso a todas as informações dos usuários registrados do sistema Blean.

Figura 15: Diagrama de navegação

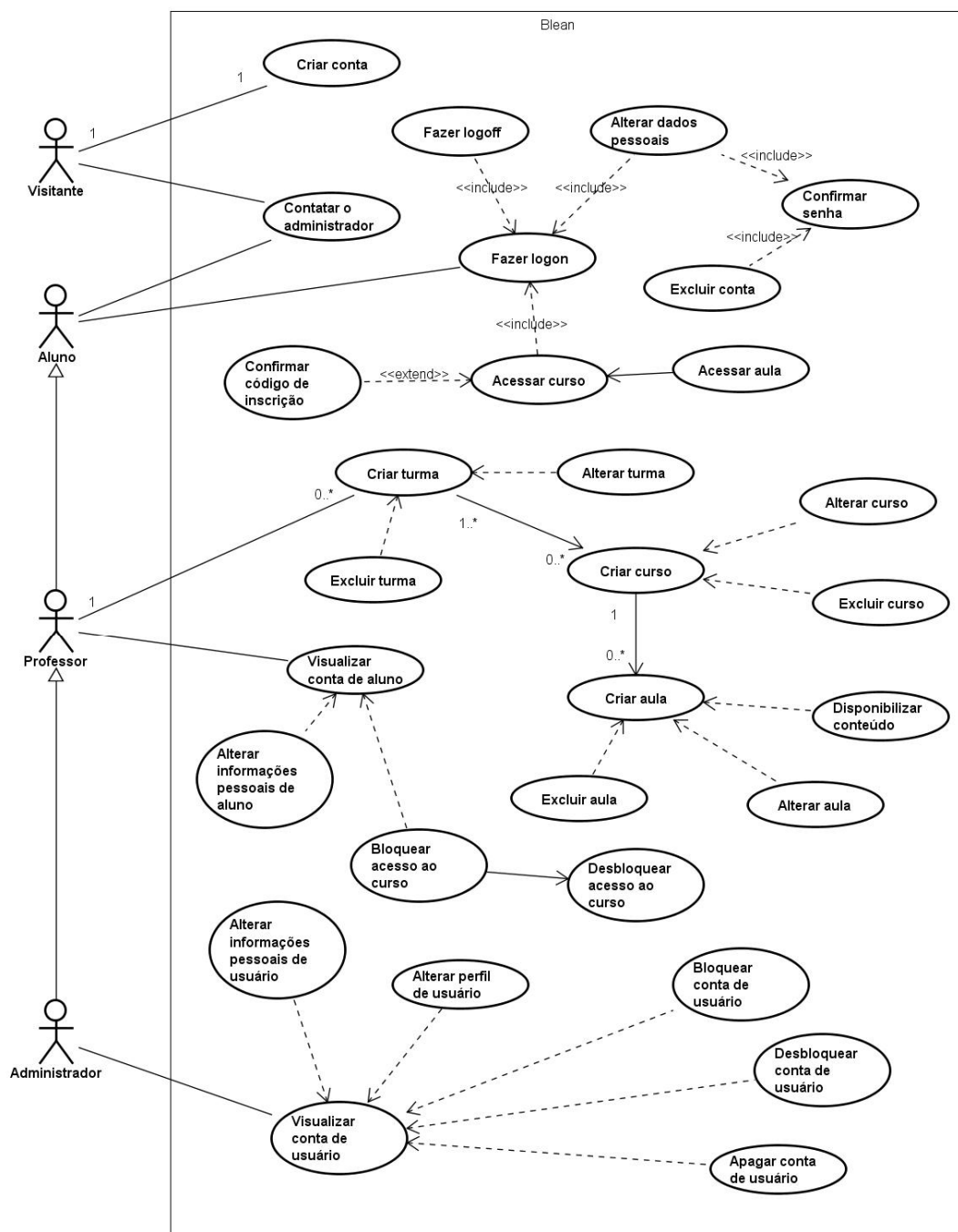


Fonte: Próprio autor (2017)

3.2.2 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso representa as ações que cada usuário pode efetuar. Pode ser obrigatória a realização de uma ação para que outra possa ser executada, por exemplo: para a ação de exclusão de uma conta, é necessária a ação de confirmação de senha do usuário. Este diagrama auxilia no desenvolvimento do diagrama de classes.

Figura 16: Diagrama de casos de uso

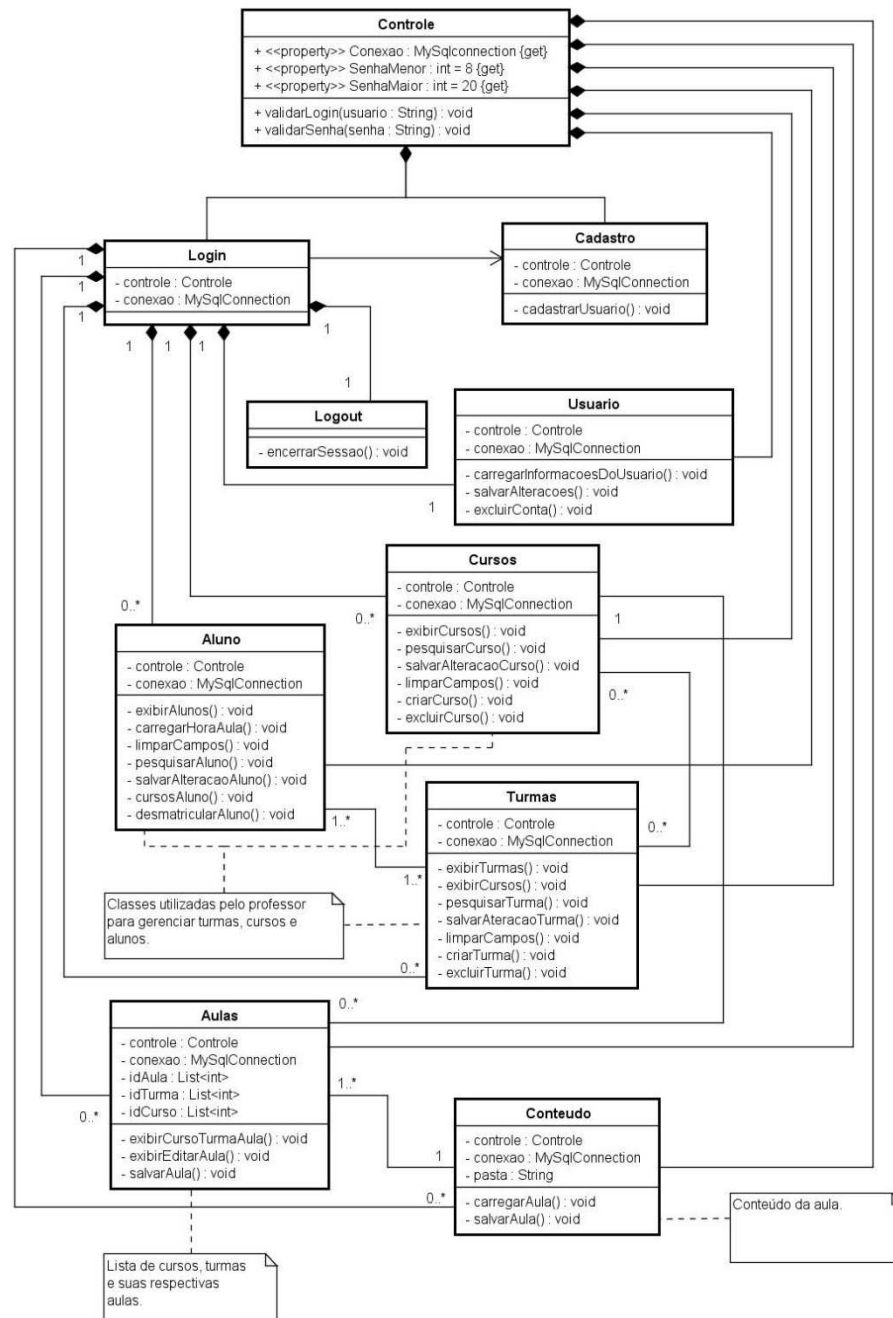


Fonte: Próprio autor (2017)

3.2.3 DIAGRAMA DE CLASSES

O diagrama de classes apresenta a estrutura de classes, seus atributos, métodos e relacionamentos. Por exemplo, a classe *Controle* relaciona-se com todas as classes, compartilhando seu atributo de conexão com o banco de dados, enquanto a classe *Logout* relaciona-se apenas com a classe *Login*, quando é requisitado o encerramento da sessão do usuário ativo no sistema.

Figura 17: Diagrama de classes



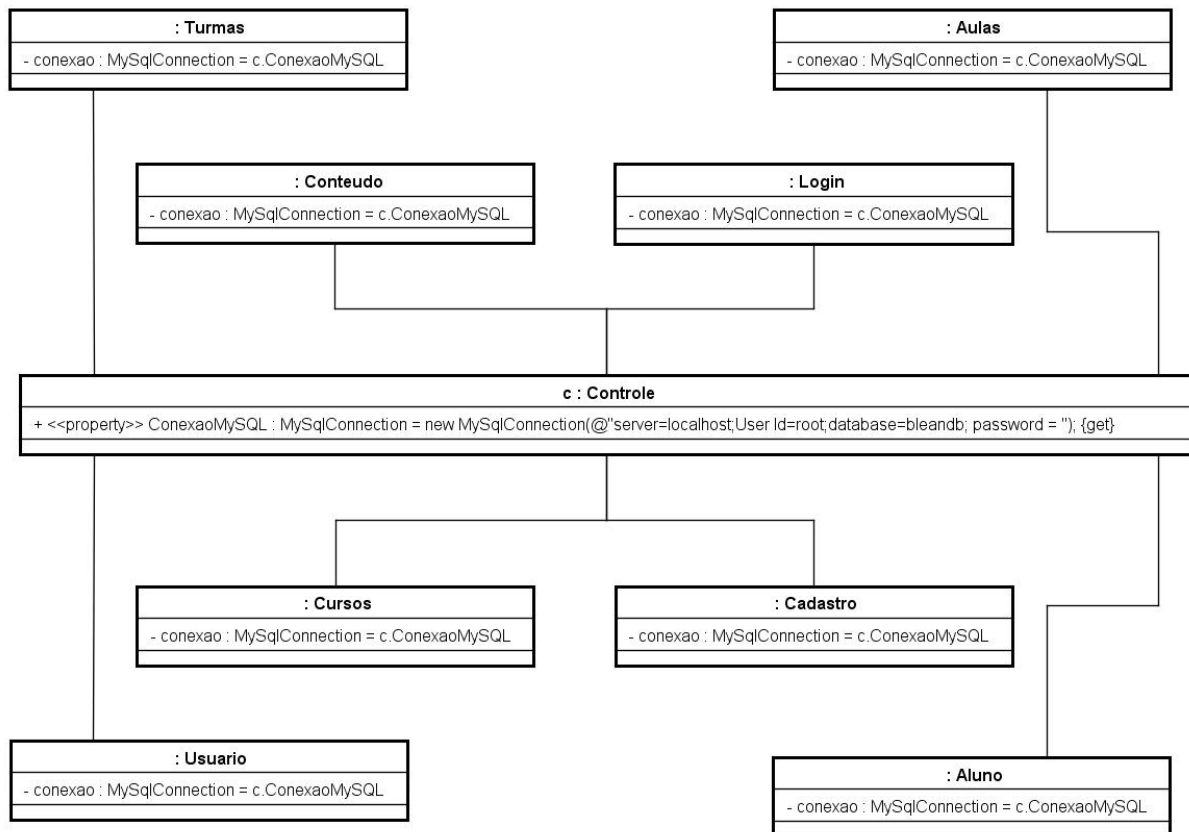
Fonte: Próprio autor (2017)

3.2.4 DIAGRAMA DE OBJETOS

O diagrama de objetos mostra a principal relação da classe Controle com as outras classes. Através do compartilhamento realizado pelo bloco de código de acesso *get* da propriedade *ConexaoMySQL*, é possível alterar a *string* de conexão apenas nesta classe, propagando-a através do objeto “c” instanciado nas outras classes. Nenhuma classe pode alterar a *string* de conexão, pois a propriedade não possui o bloco de código de acesso *set*.

Este processo é importante para agilizar e evitar erros na transferência do sistema alocado em um servidor local para o servidor web e, por qualquer outro motivo, quando houver alterações no SGBD.

Figura 18: Diagrama de objetos



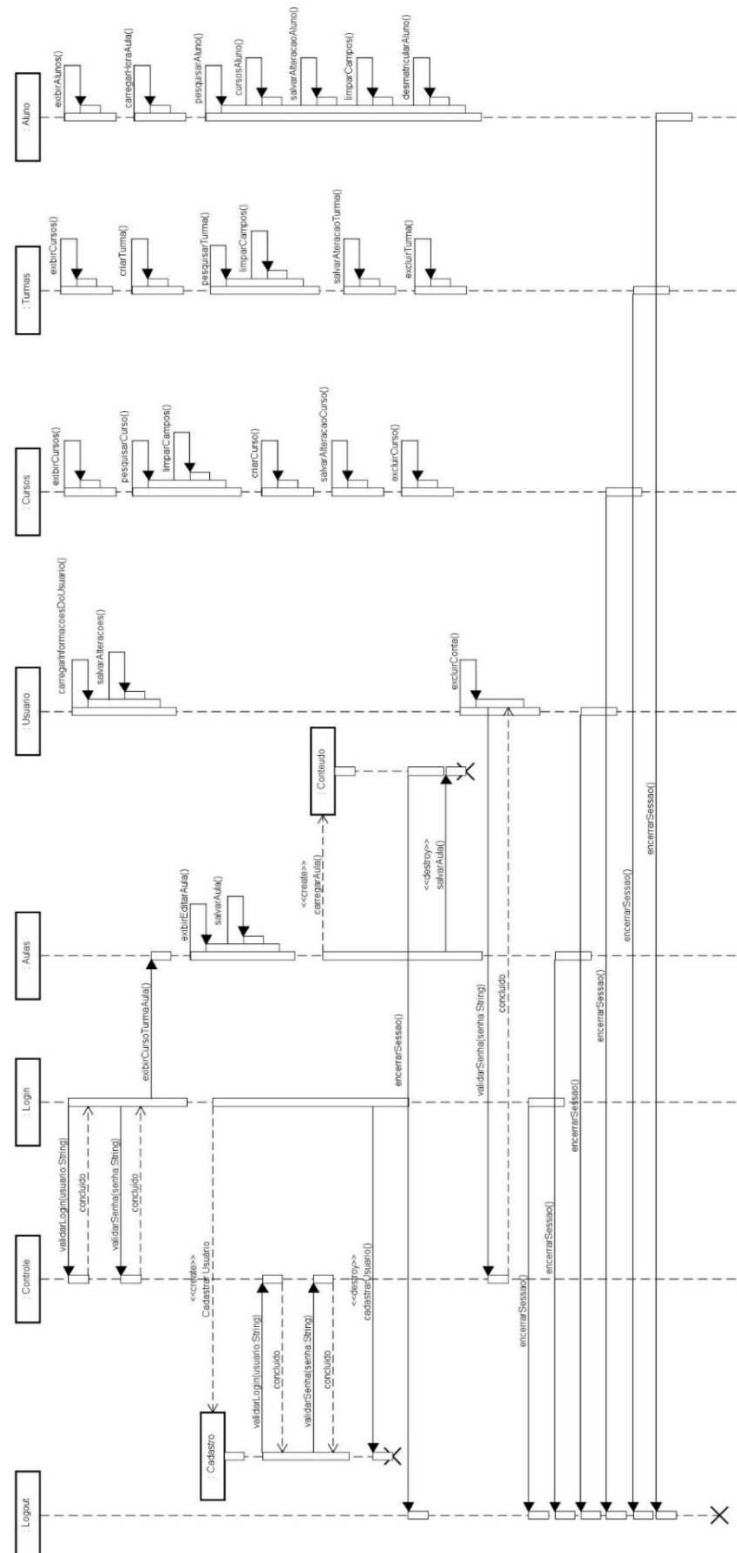
Fonte: Próprio autor (2017)

3.2.5 DIAGRAMA DE SEQUÊNCIAS

O diagrama de sequências representa um modelo estático das instâncias de classes e as interações entre elas por requisições de atributos ou métodos. Por exemplo, a classe *Login* faz a requisição de validação de dados a classe *Controle*, que retorna uma mensagem dizendo

se o *login* e senha digitados são válidos. Caso sejam inválidos, a classe *Login* avisa o usuário sobre o erro, e caso sejam válidos, ela invoca a classe *Aulas* que, faz requisições a ela mesmas para exibir, criar ou editar aulas.

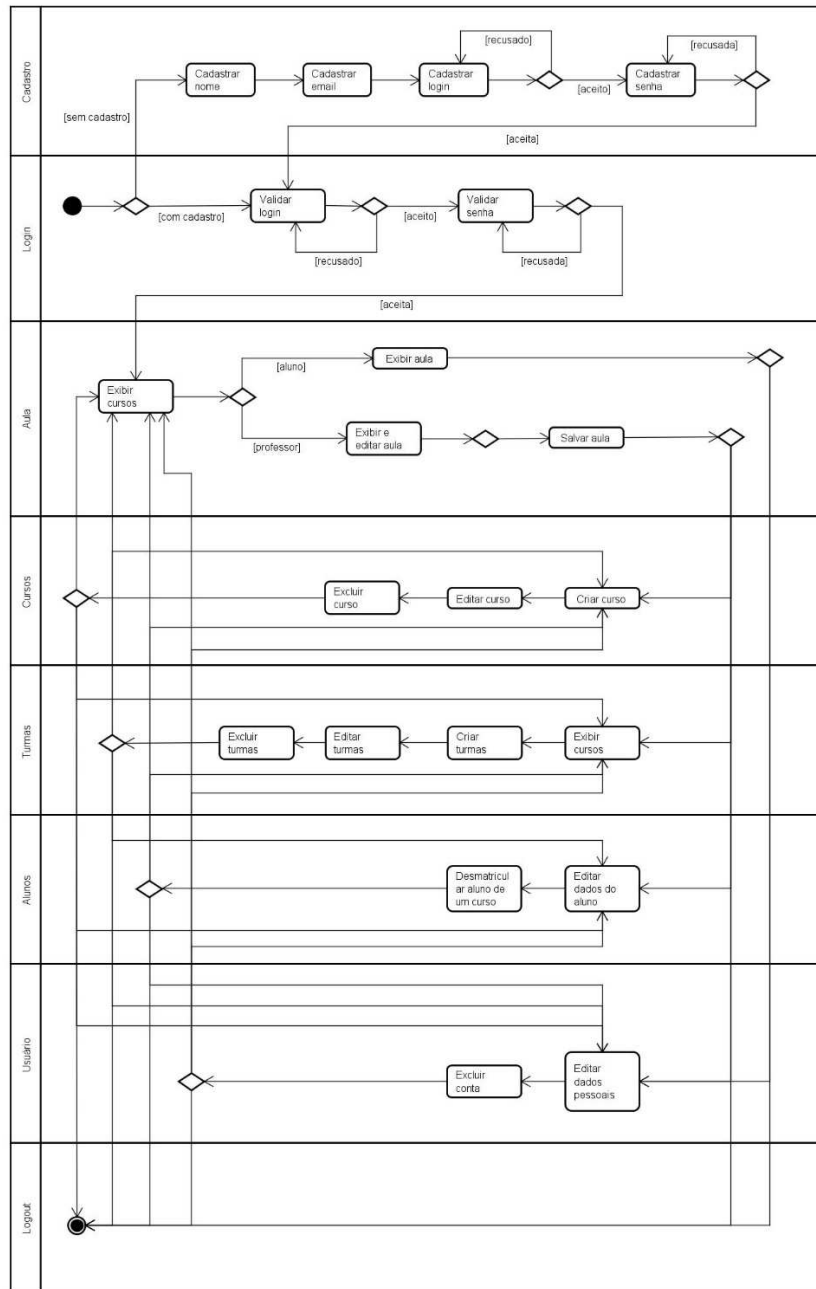
Figura 19: Diagrama de seqüências



3.2.6 DIAGRAMA DE ATIVIDADES

O diagrama de atividades representa as atividades de um objeto. Por exemplo, na classe *Login*, existe uma atividade sequencial para validação do usuário. Ao tentar acesso ao sistema, o usuário digita o nome de usuário e senha e pressiona o botão para confirmar. Então, a classe *Login* recebe o nome de usuário e faz sua validação. Se o nome de usuário não estiver no banco de dados, a classe avisa ao usuário que o nome é inválido e aguarda o usuário digitar o nome correto. Se o nome digitado existir, a classe passa a fazer a validação da senha digitada.

Figura 20: Diagrama de atividades

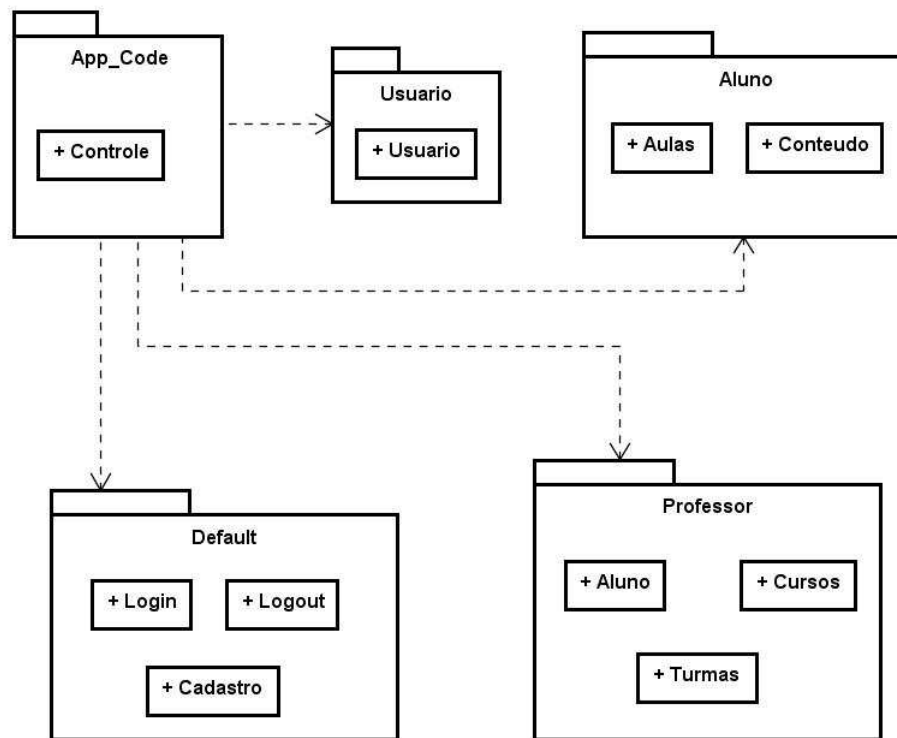


Fonte: Próprio autor (2017)

3.2.7 DIAGRAMA DE PACOTES

O diagrama de pacotes representa a organização lógica do sistema. Um pacote é uma pasta que armazena classes que podem interagir com classes do mesmo ou de outro pacote. Observe que o pacote *App_Code* comunica-se com todos os outros pacotes do sistema, já que ele possui e compartilha um atributo de conexão ao banco de dados presente em sua classe Controle.

Figura 21: Diagrama de pacotes

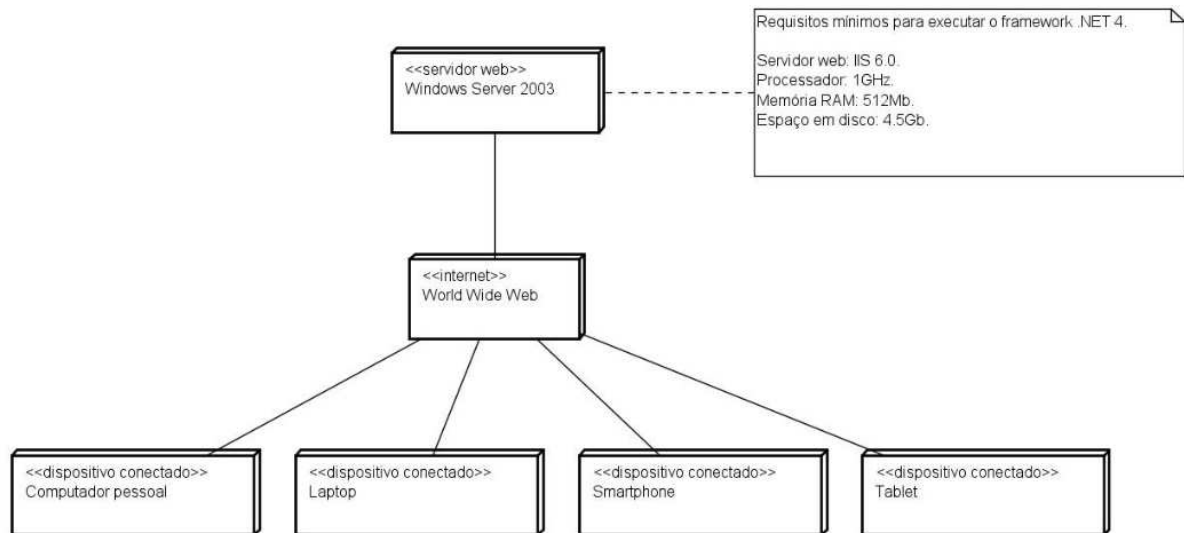


Fonte: Próprio autor (2017)

3.2.8 DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação exibe a modelagem física do sistema. O servidor necessário para sua hospedagem, o meio de distribuição e os dispositivos que poderão acessá-lo.

Figura 22: Diagrama de implantação

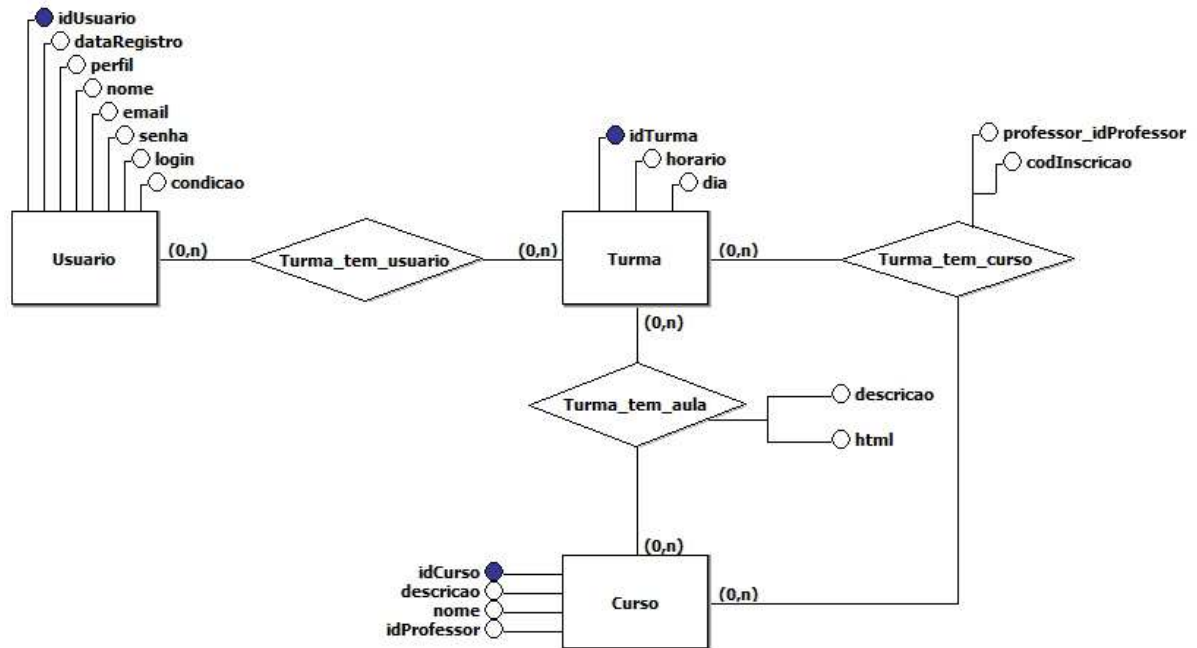


Fonte: Próprio autor (2017)

3.2.9 DIAGRAMA ENTIDADE RELACIONAMENTO (DER)

O diagrama entidade relacionamento é um modelo conceitual que exhibe as entidades com seus atributos e o relacionamento entre elas. As entidades (retângulos) e relacionamentos (losangos) se tornarão tabelas no banco de dados, enquanto os atributos (círculos) serão colunas de uma tabela. O círculo preenchido é uma chave primária da tabela, que identifica uma linha na tabela. Por exemplo, a entidade Usuario tem a chave primária idUsuario, que é única para cada usuário do sistema Blean. O texto entre parênteses é chamado de cardinalidade e representa a quantidade de relacionamentos que uma entidade pode ter, por exemplo: uma Turma contém 0 ou muitos usuários (0,n) e um Usuário pertence a 0 ou muitas Turmas (0,n).

Figura 23: DER (diagrama entidade relacionamento)

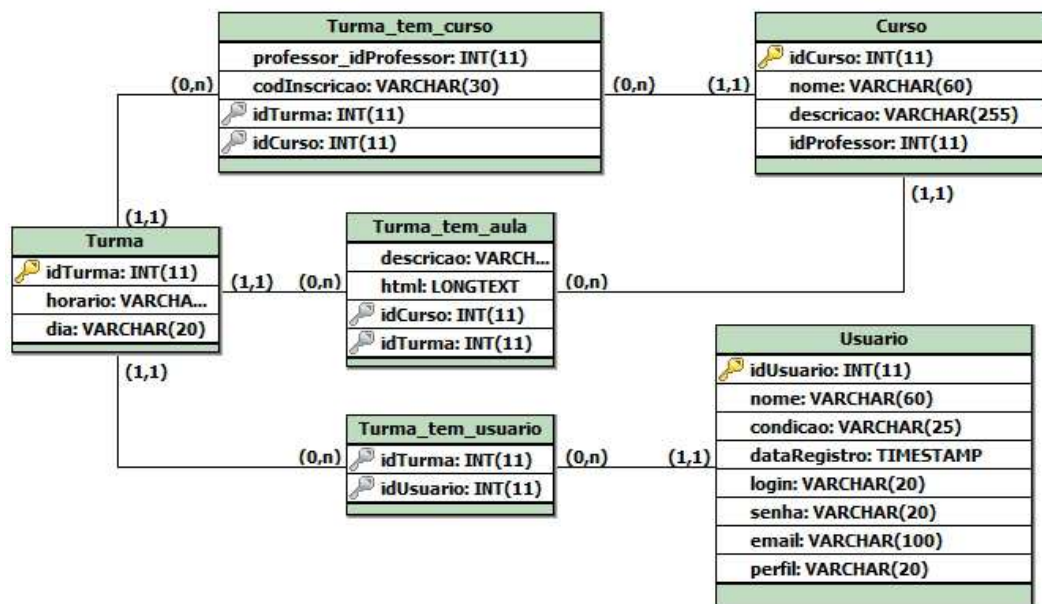


Fonte: Próprio autor (2017)

3.2.10 MODELO ENTIDADE RELACIONAMENTO (MER)

O modelo entidade relacionamento é um modelo lógico elaborado a partir do modelo conceitual e, representa as tabelas do banco de dados com seus atributos, tipos e cardinalidade. Por exemplo, a tabela Curso possui o atributo nome do tipo varchar (texto) de 60 caracteres.

Figura 24: MER (modelo entidade relacionamento)



Fonte: Próprio autor (2017)

3.2.11 DICIONÁRIO DE DADOS

O dicionário de dados descreve os campos das tabelas do banco de dados. Por exemplo, na tabela Curso, o campo nome refere-se ao nome do curso, enquanto na tabela Usuario o campo nome refere-se ao nome completo do usuário. Este dicionário é importante para a implementação e manutenção do sistema.

Tabela 2: Dicionário de dados - tabela "Curso"

Coluna	Tipo	Descrição
idCurso	int(11)	Chave primária.
nome	varchar(60)	Nome do curso.
descricao	varchar(60)	Descrição do curso.
idProfessor	int(11)	Número do professor registrado.

Fonte: Próprio autor

Tabela 3: Dicionário de dados - tabela "Turma"

Coluna	Tipo	Descrição
idTurma	int(11)	Chave primária.
dia	varchar(20)	Dia da semana.
horário	varchar(45)	Horário de aula (início e término).

Fonte: Próprio autor

Tabela 4: Dicionário de dados - tabela "Turma_tem_aula"

Coluna	Tipo	Descrição
idAula	int(11)	Chave primária.
descricao	varchar(255)	Descrição da aula.
turma_idTurma	int(11)	Chave estrangeira. Referente a idTurma.
curso_idCurso	int(11)	Chave estrangeira. Referente a idCurso.
html	longtext	Código HTML de exibição da aula.

Fonte: Próprio autor

Tabela 5: Dicionário de dados - tabela "Turma_tem_curso"

Coluna	Tipo	Descrição
turma_idTurma	int(11)	Chave primária. Referente a idTurma.
curso_idCurso	int(11)	Chave primária. Referente a idCurso.
professor_idProfessor	int(11)	Número do professor registrado.
codInscricao	varchar(30)	Código para inscrição no curso.

Fonte: Próprio autor

Tabela 6: Dicionário de dados - tabela "Usuario"

Coluna	Tipo	Descrição
idUsuario	int(11)	Chave primária.
nome	varchar(60)	Nome completo do usuário.
login	varchar(20)	Nome de acesso.
senha	varchar(20)	Senha de acesso.
email	varchar(100)	E-mail do usuário.
perfil	varchar(20)	Perfil do usuário (aluno ou professor).
status	varchar(25)	Usuário ativo ou inativo.
dataRegistro	timestamp	Data de registro do usuário.

Fonte: Próprio autor

Tabela 7: Dicionário de dados - tabela "Turma_tem_usuario"

Coluna	Tipo	Descrição
turma_idTurma	int(11)	Chave primária. Referente a idTurma.
usuario_idUsuario	int(11)	Chave primária. Referente a idUsuario.

Fonte: Próprio autor

3.3 IMPLEMENTAÇÃO

O desenvolvimento do ambiente virtual de aprendizagem Blean ocorreu em um laptop com resolução de tela de 1366 pixels de largura por 768 pixels de altura, sistema operacional Windows 10, processador Core i5 de 2,53GHz, memória RAM de 8GB e processador gráfico de 550MHz com memória dedicada de 1GB. Estes valores superam os

requisitos mínimos⁴² para execução do *framework* .NET 4, utilizado na implementação deste sistema.

Utilizou-se o IDE Visual Studio Community 2015 para desenvolvimento das páginas HTML e ASPX, scripts JavaScript, folha de estilos CSS, extensões AJAX⁴³ adicionadas através do gerenciador de extensões NuGet e, programação em linguagem C#. O Visual Studio Community 2015 possui o servidor web IIS Express, que é iniciado automaticamente quando requerida a visualização do site. Para o banco de dados, utilizou-se o SBGD MySQL, presente no servidor local Apache, disponibilizado pelo software Xampp 5.6.23.

Realizou-se testes nos navegadores:

- Microsoft Edge 38.14393.1066.0;
- Google Chrome 57.0.2987.133 (64-bit);
- Mozilla Firefox 52.0.2 (32-bit);
- Firefox Developer Edition 54.0a2 (2017-04-18) (64-bit);
- Opera 44.0.2510.1218 (PGO) e;
- Internet Explorer 11.1066.14393.0.

Os testes apresentaram resultados favoráveis quanto a tempo de resposta e renderização dos componentes das páginas exibidas nos navegadores citados.

3.4 IMAGENS DAS TELAS

O ambiente virtual de aprendizagem Blean, tem em seu menu inicial, a apresentação de três páginas. Este menu altera-se conforme a utilização e perfil de acesso do usuário.

A figura 25 exibe a tela Início, elas contêm informações sobre os recursos do sistema.

⁴² Veja os requisitos mínimos no diagrama de implantação na seção 3.2.8.

⁴³ AJAX significa *Asynchronous JavaScript e XML*. Em poucas palavras, é o uso do objeto XMLHttpRequest para se comunicar com os scripts do lado do servidor. Fonte: https://developer.mozilla.org/pt-BR/docs/AJAX/Getting_Started, acesso em 20/04/2017.

Figura 25: Perfil visitante - tela inicial



Fonte: Próprio autor

A figura 26 exibe a tela de acesso ao sistema e um link para novos usuários realizarem o cadastro.

Figura 26: Perfil visitante - login

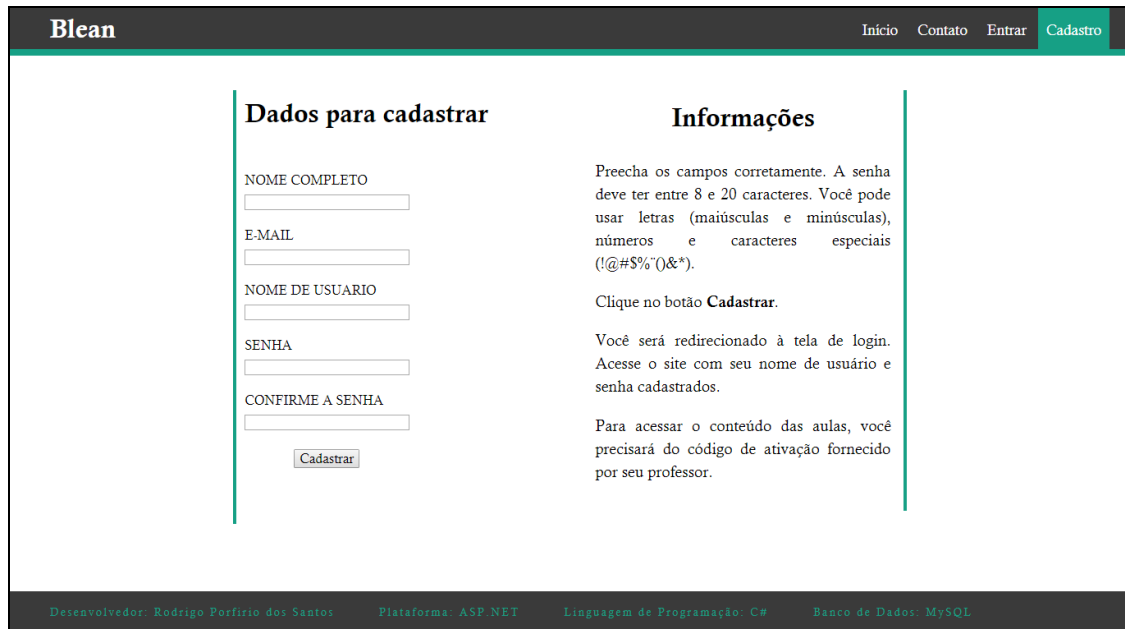


Fonte: Próprio autor

A figura 27 exibe a tela de cadastro de novos usuários e informações importantes para sua realização. O cadastro é feito com informações simples: nome completo, e-mail nome de usuário, senha e confirmação de senha. A senha deverá ter no mínimo oito caracteres e no máximo vinte caracteres, e pode conter números, letras maiúsculas, letras minúsculas e caracteres especiais como * ou \$. O nome de usuário e e-mail são únicos para cada usuário e,

a senha é criptografada, para garantir a integridade dos dados do usuário. Um novo usuário é cadastrado com o perfil de aluno, tendo que pedir ao administrador para fazer a alteração do perfil para professor.

Figura 27: Perfil visitante - cadastro de novo usuário



Blean Início Contato Entrar **Cadastro**

Dados para cadastrar

NOME COMPLETO

E-MAIL

NOME DE USUARIO

SENHA

CONFIRME A SENHA

Informações

Preecha os campos corretamente. A senha deve ter entre 8 e 20 caracteres. Você pode usar letras (maiúsculas e minúsculas), números e caracteres especiais (!@#%^(~()&*).

Clique no botão **Cadastrar**.

Você será redirecionado à tela de login. Acesse o site com seu nome de usuário e senha cadastrados.

Para acessar o conteúdo das aulas, você precisará do código de ativação fornecido por seu professor.

Desenvolvedor: Rodrigo Porfirio dos Santos Plataforma: ASP.NET Linguagem de Programação: C# Banco de Dados: MySQL

Fonte: Próprio autor

A figura 28 exibe a tela de edição de dados pessoais de um usuário registrado. Para toda e qualquer alteração de dados, o usuário deverá confirmar sua senha atual para garantir a integridade de seus dados.

Figura 28: Usuário registrado - dados pessoais

Fonte: Próprio autor

A figura 29 exibe a tela de cursos do professor. Nesta tela, o professor pode criar um novo curso ou editar um curso existente. O professor precisa criar um curso para, então, criar uma turma. Um curso pode pertencer a diversas turmas.

Figura 29: Professor - criar ou editar cursos

Curso	Descrição
AutoCAD 2D	Configurações, ferramentas de desenho e edição, desenho mecânico, desenho civil, texto, cotas, blocos e plotagem.
AutoCAD 3D	Modelagem (peças mecânicas e maquetes), projeção, plotagem, materiais, iluminação, localização geográfica, renderização e configurações avançadas de renderização.
Autodesk Inventor	Modelagem, montagem, soldagem, vista explodida, renderização, animação, desenho e plotagem.
3ds Max	Modelagem poligonal, shapes, materiais, mapeamento, iluminação e renderização (Default Scanline Renderer e Mental Ray).
CorelDRAW	Vetorização e planta humanizada.
Photoshop	Edição de imagens, efeitos, correção, vetorização, diagramação, criação de mapas para uso em software 3D, introdução a HTML e CSS.
Microsoft Word 2016	Configuração, formatação de texto, inserção de imagens, tabelas, fórmulas e expressões matemáticas.

Fonte: Próprio autor

A figura 30 exibe a tela de turma do professor. Nesta tela, o professor pode criar uma nova turma ou editar uma turma existente. Uma mesma turma pode conter diversos cursos, estes cadastrados anteriormente. Cada turma cadastrada deverá conter um código de inscrição, que será fornecido pelo professor a seus alunos para que estes possam assistir o curso específico. Se o professor alterar o código da turma, todos os alunos registrados nesta turma perderão o acesso a esta.

Figura 30: Professor - criar ou editar turmas

Dia
 Entrada
 Saída
 Curso
 Cod. inscrição

Dia	Horário	Curso	Cód. Inscrição
quinta-feira	18:30 as 20:30	Autodesk Inventor	aiv1234
quinta-feira	20:30 as 22:30	AutoCAD 3D	acd3d
sábado	08:00 as 10:00	Photoshop	phits78
sábado	10:00 as 12:00	AutoCAD 2D	acd2d65
sábado	13:00 as 15:00	Autodesk Inventor	aivd365
sábado	15:00 as 17:00	Photoshop	spf54
segunda-feira	20:00 as 21:30	Microsoft Word 2016	msw16

Desenvolvedor: Rodrigo Porfirio dos Santos Plataforma: ASP.NET Linguagem de Programação: C# Banco de Dados: MySQL

Fonte: Próprio autor

A figura 31 exibe a tela de criação de aulas do professor. Toda turma registrada aparece nesta tela para que o professor possa adicionar aulas a esta. Clicando sobre uma aula, o professor é direcionado a tela de edição de conteúdo de aula.

Figura 31: Professor – criar aula ou editar o título da aula

Curso	Dia	Horário
Microsoft Word 2016	segunda-feira	20:00 as 21:30
Photoshop	sábado	15:00 as 17:00
Descrição <input type="text"/> <input type="button" value="Adicionar Aula"/>		
003 21/04/2017 Ferramentas de seleção.		<input checked="" type="checkbox"/> <input type="checkbox"/>
002 21/04/2017 Ajustes (brilho/contraste, níveis, curvas e exposição), máscara.		<input checked="" type="checkbox"/> <input type="checkbox"/>
001 21/04/2017 Interface, padrão de cores (rgb, cmyk), pincéis, camadas, mover.		<input checked="" type="checkbox"/> <input type="checkbox"/>
Autodesk Inventor	sábado	13:00 as 15:00
AutoCAD 2D	sábado	10:00 as 12:00
Photoshop	sábado	08:00 as 10:00
AutoCAD 3D	quinta-feira	20:30 as 22:30
Autodesk Inventor	quinta-feira	18:30 as 20:30

Desenvolvedor: Rodrigo Porfirio dos Santos Plataforma: ASP.NET Linguagem de Programação: C# Banco de Dados: MySQL

Fonte: Próprio autor

A figura 32 exibe a tela de edição de conteúdo da aula do professor. O professor pode inserir texto, imagens, vídeos e slides diretamente na página da aula e disponibilizar arquivos para download.

Figura 32: Professor - criar ou editar o conteúdo da aula

Crie qualquer coisa que imaginar. Onde quer que esteja.

O melhor aplicativo de imagens e design do mundo está sempre presente em projetos criativos. Trabalhe no desktop e em dispositivos móveis para criar e aprimorar fotografias, designs da Web e de aplicativos para dispositivos móveis, ilustrações 3D, vídeos e muito mais.

Vá de uma página em branco a um trabalho incrível.

O Photoshop faz parte da Creative Cloud, o que significa que você pode acessar todos os seus ativos, inclusive o Adobe Stock, direto do aplicativo e transformar qualquer inspiração em uma linda obra de arte rapidamente. E, com a Adobe CreativeSync, tudo fica conectado entre os aplicativos para desktop e dispositivos móveis.

Fonte: <https://www.adobe.com/br/products/photoshop.html>

Desenvolvedor: Rodrigo Porfirio dos Santos Plataforma: ASP.NET Linguagem de Programação: C# Banco de Dados: MySQL

Fonte: Próprio autor

A figura 33 exibe a tela de alunos do professor. O professor pode editar informações pessoais de seus alunos, cancelar o acesso a um ou a todos os cursos, desativar ou reativar a conta de seus alunos.

Figura 33: Professor - gerenciar contas de alunos

The screenshot shows the 'Blean' web application interface. At the top, there is a navigation bar with the user name 'Olá Rodrigo!' and menu items: 'Início', 'Alunos', 'Cursos', 'Turmas', 'Aulas', 'Contato', and 'Sair'. The 'Alunos' menu is currently selected.

On the left side, there is a registration form for a student. The form includes the following fields and controls:

- ID: 8
- Nome: Claudemira Amaro
- E-mail: Nullam.nisi.Maecenas@cursumdiamat.org
- Login: aluno1
- Senha: [input field]
- Situação: ativo (dropdown menu)
- Dia: [input field]
- Horário: [input field]
- Buttons: Pesquisar, Limpar Campos, Salvar, Excluir

On the right side, there is a table listing courses with columns for 'Curso', 'Dia', and 'Horário'. Each row has a 'Desmatricular' button.

Curso	Dia	Horário
AutoCAD 3D	quinta-feira	20:30 as 22:30
Photoshop	sábado	08:00 as 10:00
Photoshop	sábado	15:00 as 17:00
Microsoft Word 2016	segunda-feira	20:00 as 21:30

At the bottom of the page, there is a footer with technical information: 'Desenvolvedor: Rodrigo Porfírio dos Santos', 'Plataforma: ASP.NET', 'Linguagem de Programação: C#', and 'Banco de Dados: MySQL'.

Fonte: Próprio autor

A figura 43 exibe a tela de turmas do aluno. Nesta tela, o aluno digita, no campo correspondente, o código de inscrição fornecido pelo professor. Se o código for válido, a turma aparece na tela e o aluno pode clicar sobre uma aula para acessar o conteúdo.

Figura 34: Aluno - inscrever-se e visualizar turmas e aulas

The screenshot shows the 'Blean' web application interface. At the top, there is a navigation bar with the user name 'Olá Claudemira!' and menu items: 'Início', 'Aulas', 'Contato', and 'Sair'. The 'Aulas' menu is currently selected.

On the left side, there is a registration form for a student. The form includes the following fields and controls:

- Código de inscrição: [input field]
- Button: Inscrever-se

On the right side, there is a table listing courses with columns for 'Curso', 'Dia', and 'Horário'. The 'Photoshop' row is highlighted in green.

Microsoft Word 2016	segunda-feira	20:00 as 21:30
Photoshop	sábado	15:00 as 17:00

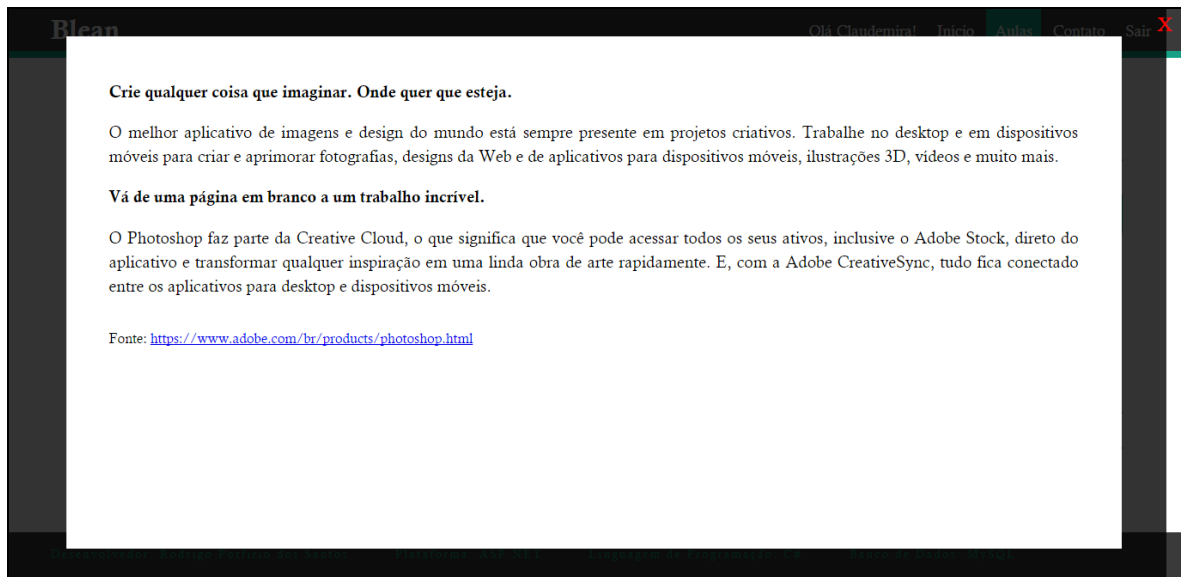
Below the table, there is a list of lessons with columns for 'ID', 'Data', and 'Conteúdo':

- 003 | 21/04/2017 | Ferramentas de seleção.
- 002 | 21/04/2017 | Ajustes (brilho/contraste, níveis, curvas e exposição), máscara.
- 001 | 21/04/2017 | Interface, padrão de cores (rgb, cmyk), pincéis, camadas, mover.

At the bottom of the page, there is a footer with technical information: 'Desenvolvedor: Rodrigo Porfírio dos Santos', 'Plataforma: ASP.NET', 'Linguagem de Programação: C#', and 'Banco de Dados: MySQL'.

Fonte: Próprio autor

A figura 35 exibe o conteúdo da aula para o aluno. Nesta tela o aluno tem acesso ao conteúdo disponibiliza por seu professor.

Figura 35: Aluno - visualizar conteúdo da aula

Fonte: Próprio autor

A figura 36 exibe a tela de usuário para o administrador. Nesta tela, o administrador gerencia as contas de alunos e professores, podendo alterar dados pessoais, desativar e ativar contas, remover o acesso dos alunos a cursos e alterar o perfil do aluno para professor.

Figura 36: Administrador: visualizar ou editar contas de usuários

Blean Olá Administrador! Início **Usuários** Turmas Cursos Aulas Contato Sair

Nome
 E-mail
 Login
 Senha
 Perfil
 Situação
 Dia
 Horário

Nome	Perfil	Email	Registro
● Rodrigo Porfirio dos Santos	professor	rodrigopcad@gmail.com	21/abril/2017
● Adosindo Torrado	professor	libero.et.tristique@leoCrasvehicula.ca	10/abril/2018
● Adélia Affonso	professor	aliquet@sit.edu	02/outubro/2017
● Wanda Newman	professor	imperdiet.dictum@malesuadamalesuada.com	31/julho/2017
● Jackson Floyd	aluno	tincidunt.nunc@uterosnon.org	06/janeiro/2018
● Colton Riley	professor	Proin@sagittislobortis.ca	02/maio/2016
● Claudemira Amaro	aluno	Nullam.nisl.Maecenas@cursusdiamat.org	13/março/2017
● Marshall Rollins	aluno	mauris.blandit.mattis@gravidamolestiearcu.net	03/fevereiro/2018
● Evan David	aluno	imperdiet.ornare@dapibus.edu	23/fevereiro/2018
● Bianca Preston	professor	auctor@commodoauctorvelit.edu	28/março/2017
● Dora Diaz	aluno	urna.Vivamus@dapibusrutrum.edu	29/dezembro/2017
● Alisa Flowers	professor	sagittis.augue.eu@purusactellus.ca	25/fevereiro/2018
● Maria Cristina	aluno	interdum.urna@sit.edu	01/fevereiro/2016

Desenvolvedor: Rodrigo Porfirio dos Santos Plataforma: ASP.NET Linguagem de Programação: C# Banco de Dados: MySQL

Fonte: Próprio autor

3.5 RESULTADO

O sistema Blean foi desenvolvido para oferecer facilidade de uso para professores e alunos. As telas do sistema são limpas, sem excesso de informações. O menu superior, exhibe em destaque a tela ativa e facilita a navegação, pois cada item possui identificação coerente com sua finalidade. O professor não terá trabalho para matricular alunos em suas turmas, já que o aluno fica encarregado de fazê-lo através do código de inscrição fornecido pelo professor em sala de aula ou por e-mail, já o trancamento fica a cargo do professor, bastando alterar o código de inscrição da turma para trancar a matrícula de todos.

As senhas são criptografadas e os campos de inserção de texto passam por validações de segurança, afim de prevenir ataques e garantir a segurança de dados.

4 CONSIDERAÇÕES FINAIS

O desenvolvimento de um sistema não é uma tarefa simples. É necessário um estudo elaborado da finalidade da aplicação, necessidade dos utilizadores, análise de tecnologias disponíveis e conhecimento técnico dos desenvolvedores, modelagem do sistema e banco de dados, escolha de linguagem de programação, banco de dados e plataforma de desenvolvimento Web, implementação, testes, revisões na modelagem, implementação e mais testes. Além disso, para que o desenvolvimento do ambiente virtual de aprendizagem apresentado neste trabalho pudesse ser realizado, passaram-se 289 anos de criações e evolução de metodologias e tecnologias, com início em 1728 quando o professor Caleb Phillips anunciou em um jornal seu curso de taquigrafia, o primeiro curso a distância registrado.

Por estas razões, o resultado deste trabalho é o trabalho de muita gente: do graduando, que teve a ideia, trabalho de pesquisa e desenvolvimento; dos professores, que compartilharam conhecimento em sala de aula, nos corredores da instituição de ensino e por e-mail; dos funcionários da biblioteca da instituição de ensino, que organizam os livros, auxiliam na pesquisa de títulos e retiram os livros de suas prateleiras e os entregam em mãos e; de cada pessoa, grupo de trabalho e empresa que contribuíram para a criação das metodologias e tecnologias descritas neste documento.

O objetivo deste projeto foi alcançado. A pesquisa para o embasamento teórico foi a parte mais demorada. O desenvolvimento ocorreu com tranquilidade, com algumas partes de implementação programática da tela de exibição de aulas ocorrendo com as seguintes dificuldades: desenvolvimento de instrução SQL⁴⁴, pela complexidade desta, que envolveu a consulta simultânea em quatro tabelas do banco de dados e criação de condições para sua montagem, separando administrador, professor e aluno e; na montagem da exibição de aulas⁴⁵, que necessitou da implementação de recursos de imagens, classes CSS, atributos de *tag* HTML, componentes Ajax e um laço de repetição bem estruturado para exibir as aulas na ordem correta. Mas as dificuldades foram solucionadas e a implementação ocorreu conforme a modelagem do sistema.

O desenvolvimento do ambiente virtual de aprendizagem Blean não finaliza aqui. Futuras melhorias no código fonte e implementações de funcionalidades serão realizadas.

⁴⁴ Apêndice A – Instrução SQL (código fonte).

⁴⁵ Apêndice B – Exibição de aulas (código fonte).

APÊNDICE A – INSTRUÇÃO SQL (CÓDIGO FONTE)

```
string sql = "SELECT c.idCurso, c.nome, c.idProfessor, t.idTurma, t.dia, t.horario FROM
curso c ";
sql += " JOIN turma_tem_curso ttc";
sql += " ON c.idCurso = ttc.curso_idCurso";
sql += " JOIN turma t";
sql += " ON t.idTurma = ttc.turma_idTurma";

if (!Session["idUsuario"].ToString().Equals("1") &&
Session["perfil"].Equals("professor"))
    sql += " WHERE c.idProfessor = " + Session["idUsuario"];
if (Session["perfil"].Equals("aluno"))
{
    sql += " JOIN turma_tem_usuario ttu";
    sql += " ON t.idTurma = ttu.turma_idTurma";
    sql += " WHERE ttu.usuario_idUsuario = " + Session["idUsuario"];
}
sql += " ORDER BY t.idTurma DESC";
```

APÊNDICE B – EXIBIÇÃO DE AULAS (CÓDIGO FONTE)

```

for (int j = 1; j <= paineis; j++)
{
    string numeroPainel = string.Format("{0:000} &nbsp;|&nbsp;&nbsp; |&nbsp;  |&nbsp;  |&nbsp;  |&nbsp;  |&nbsp;  |&nbsp;  |", paineis - j + 1);
    LinkButton linkButton = new LinkButton();
    Literal txtLinkButton = new Literal();
    txtLinkButton.Text = numeroPainel + descricao.ElementAt(indiceIdAula);
    linkButton.CssClass = "aula";
    linkButton.CommandArgument = lstIdTurma.ElementAt(i) + "," +
    lstIdCurso.ElementAt(i) + "," + j + "," + lstidAula.ElementAt(indiceIdAula);
    linkButton.ID = "linkButton" + lstidAula.ElementAt(indiceIdAula) +
    lstIdTurma.ElementAt(i) + lstIdCurso.ElementAt(i) + j;
    linkButton.CommandName = "EDITAR_AULA";

    if (Session["perfil"].ToString().Equals("professor"))
        linkButton.ToolTip = "Editar o conteúdo da aula.";
    else
        linkButton.ToolTip = "Visualizar aula.";

    AccordionPane acPane = (AccordionPane)accCursos.FindControl("pane" +
    lstIdCurso.ElementAt(i) + lstIdTurma.ElementAt(i));
    acPane.ContentContainer.Controls.Add(linkButton);
    linkButton.Controls.Add(txtLinkButton);

    if (Session["perfil"].ToString().Equals("professor"))
    {
        ImageButton btnEditarDescricao = new ImageButton();
        btnEditarDescricao.ImageUrl = @"~/Imagens/edit.png";
        btnEditarDescricao.ToolTip = "Editar a descrição da aula.";
        btnEditarDescricao.CssClass = "margemEsquerda";
        btnEditarDescricao.Attributes.Add("onmouseout", "this.src='../Imagens/edit.png'");
        btnEditarDescricao.Attributes.Add("onmouseover",
        "this.src='../Imagens/edit_hover.png'");
        btnEditarDescricao.CommandName = "EDITAR_DESCRICAO";
        ImageButton btnDelAula = new ImageButton();
        btnDelAula.ImageUrl = @"~/Imagens/delete.png";
        btnDelAula.ToolTip = "Deletar a aula.";
        btnDelAula.CssClass = "margemEsquerda";
        btnDelAula.Attributes.Add("onmouseout", "this.src='../Imagens/delete.png'");
        btnDelAula.Attributes.Add("onmouseover",
        "this.src='../Imagens/delete_hover.png'");
        btnDelAula.CommandName = "DELETAR_AULA";
        linkButton.Controls.Add(btnEditarDescricao);
        linkButton.Controls.Add(btnDelAula);
    }
    indiceIdAula++;
}

```


REFERÊNCIAS

.NET Framework Versions and Dependencies. 2017. *In.*: **Microsoft Developer Network**. Disponível em: <[https://msdn.microsoft.com/en-us/library/bb822049\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb822049(v=vs.110).aspx)>. Acesso em: 01 maio 2017, às 23h36min.

ALVES, João Roberto Moreira. (Rio de Janeiro). **A História da Educação a Distância no Brasil**. 2007. Disponível em: <http://www.ipae.com.br/pub/pt/cme/cme_82/index.htm>. Acesso em: 02 nov. 2016, às 10h17min.

ASP.NET e Visual Studio. 2017. *In.*: **Microsoft Developer Network**. Disponível em <[https://msdn.microsoft.com/pt-br/library/dd566231\(v=vs.110\).aspx](https://msdn.microsoft.com/pt-br/library/dd566231(v=vs.110).aspx)>. Acesso em: 09 maio 2017, às 15h05min.

ASP.NET Web Pages (Razor) FAQ. (2014). *In.*: **Microsoft**. Disponível em: <<https://docs.microsoft.com/en-us/aspnet/web-pages/overview/getting-started/aspnet-web-pages-razor-faq>>. Acesso em: 10 maio 2017, às 11h08min.

BELMONT, Vanessa; GROSSI, Márcia Gorett Ribeiro (ORG.). **Ambientes Virtuais de Aprendizagem** : um panorama da produção nacional : CEFET - MG. Belo Horizonte - MG, 2010. 10p.

BOOCH, Grady; RUMBAUGH, James e; JACOBSON, Ivar. **UML** : guia do usuário. 2. Ed. rev. e atual. Rio de Janeiro : Elsevier, 2005. 474 p. Tradução de Fábio Freitas da Silva e Cristina de Amorim Machado.

BRASIL. Constituição (1988). Lei nº 191-A, de 5 de outubro de 1988. **Constituição da República Federativa do Brasil**. Brasília, DF, 5 out. 1988. Seção 1, p. 23. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/DOUconstituicao88.pdf>. Acesso em: 11 nov. 2016, às 20h30min.

BRASIL. Constituição (2005). Decreto nº 5622, de 19 de dezembro de 2005. **Diário Oficial**. Brasília, DF, 20 dez. 2005.

CARVALHO, Marcelo Sávio Revoredo Menezes de. **A Trajetória da Internet no Brasil** : do surgimento das redes de computadores à instituição dos mecanismos de governança. 2006. 239 f. Dissertação (Mestrado em Engenharia de Sistemas e Computação) Universidade Federal do Rio de Janeiro – Departamento de Pós-Graduação de Engenharia, Rio de Janeiro, 2006. Cap. 3. Disponível em: <<http://tele.sj.ifsc.edu.br/~tisemp/RES/Internet-BR-Dissertacao.pdf>>. Acesso em: 25 nov. 2006m às 13h46min.

CERN Accelerating Science. 2017. Disponível em: <<http://home.cern>>. Acesso em: 22 maio 2017, às 00h07min.

COMMON acronyms. 2013. *In.*: **Moodle**. Disponível em: <https://docs.moodle.org/25/en/Common_acronyms>. Acesso em: 11 jan. 2017, às 14h03min.

COMO surgiu o EAD? 2016. *In.*: **Ead - Conquiste seu diploma**, 2016. Disponível em: <<http://www.ead.com.br/ead/como-surgiu-ensino-a-distancia.html>>. Acesso em: 21 out. 2016, às 23h18min.

CONNECTIVITY Visualizations. 2017. *In.*: **Akamai Technologies**. Disponível em: <<https://www.akamai.com/us/en/our-thinking/state-of-the-internet-report/state-of-the-internet-connectivity-visualization.jsp>>. Acesso em: 05 jan. 2017, às 02h27min.

DEITEL, H. M. *et. al.* **C# : como programar**. São Paulo : Pearson Education do Brasil Ltda, 2003. 1200 p. Tradução de João Eduardo Nóbrega Tortello.

EDMODO. 2016. Disponível em: <<https://www.edmodo.com>>. Acesso em: 16 jan. 2017, às 10h32min.

FILIPPO, Denise Del Re. SZTAJNBERG Alexandre. **Bem-vindo à Internet**. Rio de Janeiro : Brasport, 1996. 400p.

GETTING Started with the .NET Framework. 2017. *In.*: **Microsoft Developer Network**. Disponível em: <<https://msdn.microsoft.com/en-us>>. Acesso em: 01 maio 2017, às 20h57min.

GLOSSÁRIO do Visual Studio e do .NET Framework. 2017. *In.*: **Microsoft Developer Network**. Disponível em: <[https://msdn.microsoft.com/pt-br/library/6c701b8w\(v=vs.90\).aspx](https://msdn.microsoft.com/pt-br/library/6c701b8w(v=vs.90).aspx)>. Acesso em: 09 maio 2017, às 23h57min.

HISTÓRIA da Internet no Brasil. 2014. *In.*: **Portal Educação**, 2016 Disponível em: <<https://www.portaleducacao.com.br/informatica/artigos/53793/historia-da-internet-no-brasil>>. Acesso em: 10 nov. 2016, às 00h15min.

HISTÓRICO. [2017?]. *In.*: **TelEduc**. Disponível em: <<http://www.teleduc.org.br/?q=histórico>>. Acesso em: 12 jan. 2017, às 16h 35min.

HISTORY of Distance Learning. [2011]. *In.*: **Adult Learning Activities** : california distance learning project, 2005–2011 Disponível em: <<http://www.cdlponline.org/index.cfm?fuseaction=whatis&pg=3>>. Acesso em: 31 out. 2016, às 02h23min.

HISTORY. 2015. *In.*: **Moodle**. Disponível em: <<https://docs.moodle.org/32/en/History>>. Acesso em: 11 jan. 2017, às 15h47min.

INSTALLATION quick guide. 2016. *In.*: **Moodle**. Disponível em: <https://docs.moodle.org/32/en/Installation_quick_guide>. Acesso em: 11 jan. 2017, às 13h50min.

INTRODUCTION to the C# Language and the .NET Framework. 2016. *In.*: **Microsoft Developer Network**. Disponível em: <<https://msdn.microsoft.com/pt-br/library/z1zx9t92.aspx>>. Acesso em: 08 fev. 2017, às 18h11min.

MOODLE Statistics. [2017?]. *In.*: **Moodle**. Disponível em: <<https://moodle.net/stats>>. Acesso em: 25 maio 2017, às 11h19min.

MOODLE. [2017?]. Disponível em: <<https://moodle.org>>. Acesso em: 11 jan. 2017, às 13h45min.

O QUE é a Tecnologia Java e porque preciso dela? [2017?]. *In: Java*. Disponível em: <https://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em: 07 fev. 2017, às 20h15min.

OPEN Source Initiative. [1998?]. Disponível em: <<https://opensource.org/osd-annotated>>. Acesso em: 11 jan. 2017, às 18h23min.

ORACLE Corporation. 2017. **MySQL**. Disponível em <<https://dev.mysql.com>>. Acesso em: 23 maio 2017, às 00h35min.

QUEM Usa. [2017?]. *In.: TelEduc*. Disponível em: <<http://www.teleduc.org.br/?q=quem-usa>>. Acesso em: 25 maio 2017, às 11h29min.

RICARTE, Ivan Luiz Marques. **Programação Orientada a Objetos: uma abordagem com Java**. Campinas/SP : Universidade Estadual de Campinas/UNICAMP, Departamento de Engenharia de Computação e Automação Industrial, Faculdade de Engenharia Elétrica e de Computação. 2001. 117p. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>>. Acesso em: 17 jan. 2017, às 15h49min

SEBESTA, Robert W. **Conceitos de Linguagem de Programação**. 9. ed. Porto Alegre/RS : Bookman Companhia Editora Ltda, 2010. 792 p. Tradução de Eduardo Kessler Piveta.

SOBRE Edmodo. 2016. *In.: Edmodo*. Disponível em: <<https://www.edmodo.com/about>>. Acesso em: 25 maio 2017, às 11h35min.

TANENBAUM, Andrew S.. **Redes de Computadores**. 4. ed. Rio de Janeiro: Campus Elsevier, 2003. 632 p. Tradução de Vandenberg D. de Souza.

TELEDUC. [2017?]. Disponível em: <<http://www.teleduc.org.br>>. Acesso em: 12 jan. 2017, às 16h32min.

W3C. 2017. Disponível em: <<https://www.w3.org>>. Acesso em: 16 maio 2017, às 22h15min.