UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

PAULA RODRIGUES MADEIRA
THALIA FERREIRA PINTO

Sistema Web de Avaliação de Disciplinas

Prof. Filipe Braida do Carmo, D.Sc.
Orientador

Nova Iguaçu, Julho de 2023

Sistema Web de Avaliação de Disciplinas

Paula Rodrigues Madeira

Thalia Ferreira Pinto

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

_____

Paula Rodrigues Madeira

_____

Thalia Ferreira Pinto

Aprovado por:

_____

Prof. Filipe Braida do Carmo, D.Sc.

_____

Prof. Bruno José Dembogurski, D.Sc.

_____

Prof. Leandro Guimaraes Marques Alvim, D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Julho de 2023

Emitido em 31/07/2023

DOCUMENTOS COMPROBATÓRIOS Nº 13578/2023 - CoordCGCC (12.28.01.00.00.98)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 07/08/2023 20:21)
BRUNO JOSE DEMBOGURSKI
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###249#4

(Assinado digitalmente em 04/08/2023 21:27)
FILIPE BRAIDA DO CARMO
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###295#4

(Assinado digitalmente em 07/08/2023 17:23)
LEANDRO GUIMARAES MARQUES ALVIM
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###008#2

(Assinado digitalmente em 05/08/2023 10:19)
PAULA RODRIGUES MADEIRA
DISCENTE
Matrícula: 2016#####8

(Assinado digitalmente em 05/08/2023 10:40)
THALIA FERREIRA PINTO
DISCENTE
Matrícula: 2016#####7

Visualize o documento original em https://sipac.ufrrj.br/documentos/ informando seu número: **13578**, ano: **2023**, tipo: **DOCUMENTOS COMPROBATÓRIOS**, data de emissão: **04/08/2023** e o código de verificação: **d1451f1335**

Agradecimentos

Paula Rodrigues Madeira

Primeiramente, agradeço à toda a minha família. Principalmente aos meus pais, Paulo e Denise, que sempre fizeram tudo por mim e pela minha irmã e por nossa educação. Todo amor, apoio e acolhimento foi fundamental durante este processo.

Muito obrigada à minha irmã, Débora, por cada momento dessa vida. É muito bom saber que tenho com quem contar independente de tudo. Nossa relação é o que tenho de mais precioso.

Meus agradecimentos especiais às minhas amigas desde o início da faculdade, Thalia Ferreira e Mariana Mendes. À Thalia por ter embarcado nesse projeto comigo e por ter sido a melhor parceira de projeto final que eu poderia ter, e à Mariana Mendes, que sempre foi um exemplo de dedicação e esforço para mim.

Ao meu namorado, Gabriel Dias, por ser tão compreensivo e por dividir os problemas comigo, tornando os dias mais leves.

Por fim, agradeço a todos os professores que já tive. Em especial ao corpo docente do Departamento de Ciência da Computação do Instituto Multidisciplinar da UFRRJ, que me ajudou a trilhar o caminho até aqui. Principalmente ao professor Filipe Braida, cuja didática, incentivo e orientação tornaram esse processo possível.

Muito obrigada a todos.

Thalia Ferreira Pinto

Agradeço, primeiramente, a toda a minha família, mas especialmente aos meus pais, pelo amor, incentivo e apoio que recebi durante toda minha vida. Tudo o que sou e tudo o que tenho devo a vocês.

Também gostaria de agradecer às minhas amigas Paula Madeira e Mariana Mendes pela parceria desde o primeiro dia em que estivemos na faculdade, mas especialmente à Paula, minha dupla incrível, por sua dedicação durante todo o desenvolvimento desta monografia.

Ao meu orientador, Filipe Braida, por sempre estar disponível para nos ajudar e aconselhar ao longo do desenvolvimento deste projeto final. Obrigada por sempre nos incentivar a continuar.

Às minhas amigas, Ana Paula, Bárbara, Marina e Mayara, pela amizade e por sempre ajudarem a renovar minhas energias.

E, por fim, agradeço a todo o corpo docente do curso de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro. Obrigada por compartilharem tanto conhecimento enriquecedor e por contribuírem para que eu me tornasse a profissional que sou hoje.

Obrigada a todos.

RESUMO

Sistema Web de Avaliação de Disciplinas

Paula Rodrigues Madeira e Thalia Ferreira Pinto

Julho/2023

Orientador: Filipe Braida do Carmo, D.Sc.

Com o objetivo de fornecer uma plataforma eficiente e intuitiva para que os alunos possam avaliar suas experiências durante o período letivo, desenvolvemos um sistema web de avaliação de disciplinas. A expectativa é promover a transparência, incentivar a troca de informações entre os discentes e auxiliar na construção de uma comunidade acadêmica mais engajada e capacitada. Além de permitir que os alunos avaliem aspectos das disciplinas cursadas e dos professores que as ministraram, o sistema foi projetado para os gestores acompanharem as métricas das avaliações de todo o departamento. Esperamos que o monitoramento desses dados possa auxiliar a instituição a identificar tendências e áreas que necessitam de aprimoramento a fim de implementar ações corretivas, ajustar abordagens pedagógicas e alocar recursos de forma mais eficiente.

ABSTRACT

Sistema Web de Avaliação de Disciplinas

Paula Rodrigues Madeira and Thalia Ferreira Pinto

Julho/2023

Advisor: Filipe Braida do Carmo, D.Sc.

With the aim of providing an efficient and intuitive platform for students to evaluate their experiences during the academic period, we have developed a web system for courses evaluation. The expectation is to promote transparency, encourage information exchange among students, and contribute to building a more engaged and capable academic community. In addition to allowing students to assess various aspects of the courses they have taken and the professors who taught them, the system has been designed to enable administrators to monitor evaluation metrics across the entire department. We hope that the monitoring of these data can assist the institution in identifying trends and areas that require improvement, leading to the implementation of corrective actions, adjustments in pedagogical approaches, and more efficient allocation of resources.

Lista de Figuras

Lista de Abreviaturas e Siglas

ARPA Agência de Projetos de Pesquisa Avançada

ARPANET *Advanced Research Projects Agency Network*

CSS *Cascading Style Sheets*

ER Entidade-Relacionamento

HTML *Hypertext Markup Language*

HTTP Protocolo de Transferência de Hipertexto

MVC *Model-View-Controller*

NPM *Node Package Manager*

ORM *Object-Relational Mapping*

RN Regra de Negócio

RF Requisito Funcional

SINAES Sistema Nacional de Avaliação da Educação Superior

UFRRJ Universidade Federal Rural do Rio de Janeiro

UFF Universidade Federal Fluminense

UC Caso de Uso

UCLA Universidade da Califórnia, Los Angeles

SRI Instituto de Pesquisa de Stanford

USP Universidade de São Paulo

W3C *World Wide Web Consortium*

TCP Transmission Control Protocol

IP Internet Protocool

SIGAA Sistema Integrado de Gestão de Atividades Acadêmicas

WWW WorldWideWeb

Sumário

Capítulo 1

Introdução

De acordo com Moore e Kuol (2005), os esforços para medir o desempenho dos docentes e analisar essas medidas no contexto do desenvolvimento profissional contribuem para criar uma maior equidade de prestígio entre os componentes de ensino e pesquisa das universidades.

Ao disponibilizar uma forma de avaliar disciplinas na universidade é possível obter diversos benefícios. Primeiramente, essa avaliação permite identificar áreas que necessitam de melhorias, seja em relação ao conteúdo ou a metologia de ensino. Com base na opinião dos estudantes, o corpo docente poderá tomar decisões que visam aprimorar a qualidade de ensino.

Além disso, ao garantir que os alunos possam avaliar anonimamente as disciplinas cursadas e que essas avaliações sejam consideradas para tomadas de decisões, os estudantes podem se tornar participantes ativos e engajados do ambiente acadêmico.

Tradicionalmente, essas avaliações eram realizadas através de formulários em papel, o que limitava a eficiência e a precisão dos resultados obtidos. Porém, com o crescente uso da tecnologia pelas universidades no Brasil e no mundo, torna-se evidente a necessidade de um sistema automatizado para coletar e analisar as informações geradas pelas avaliações.

Diante desse cenário, surge a motivação para o desenvolvimento de um sistema de

avaliações de disciplinas. Com a implementação desse sistema, será possível agilizar o processo de análise dos *feedbacks* dos estudantes, facilitando a tomada de decisões pelos docentes e aprimorando a qualidade do ensino oferecido.

1.1 Objetivo

O objetivo desse trabalho consiste em desenvolver um sistema web de avaliação de disciplinas do curso de Ciência de Computação da Universidade Federal Rural do Rio de Janeiro (UFRRJ), que pode ser acessado pelo corpo discente e docente do curso.

Através da aplicação, os alunos podem avaliar as disciplinas cursadas, os professores podem visualizar as médias de avaliações recebidas pelas suas disciplinas lecionadas, e os coordenadores do curso podem analisar as médias de avaliações recebidas pelos professores.

1.2 Organização do Trabalho

Esta monografia está organizada da seguinte maneira:

- **Capítulo 1** Consiste em uma breve explicação sobre o contexto, a motivação e a proposta.
- **Capítulo 2** Nesse capítulo, apresentaremos uma contextualização sobre a Internet e discutiremos sobre as principais linguagens utilizadas no desenvolvimento de aplicações web.
- **Capítulo 3** Neste capítulo, exploraremos a motivação por trás da proposta, os trabalhos relacionados, e a modelagem da proposta, descrevendo suas características e funcionalidades.
- **Capítulo 4** Neste capítulo, abordaremos a implementação da aplicação, descrevendo as tecnologias utilizadas e a arquitetura do sistema.

- **Capítulo 5** Este capítulo consistirá em uma breve conclusão, onde apresentaremos as considerações finais sobre a proposta desenvolvida, discutiremos suas limitações e as possíveis direções para trabalhos futuros.

Capítulo 2

Fundamentação

A Internet é a base fundamental na qual os sistemas web são construídos e operam, portanto, conhecer seus princípios subjacentes é essencial para um compreendimento satisfatório. Dessa forma, esse capítulo realiza uma breve contextualização sobre o que conhecemos hoje como Internet na seção 2.1, explicando seus conceitos e nascimento. Enquanto na seção 2.2 apresentamos e descrevemos os componentes essenciais para o desenvolvimento para web.

2.1 Internet

Nos dias de hoje, entendemos a Internet como uma rede global de computadores interconectados que permite a troca de informações em tempo real e comunicação entre usuários em diferentes partes do mundo. Ela consiste em uma vasta infraestrutura física e lógica composta por milhões de dispositivos, como computadores, servidores, roteadores, cabos de fibra ótica e até mesmo satélites.

Para os cientistas Krol e Hoffman (1993), a Internet podia ser compreendida como um conjunto de protocolos comuns, como uma infraestrutura física composta por roteadores e circuitos ou mesmo como um meio de interconexão e intercomunicação. E ainda incluem as características de ser uma rede composta de redes baseadas nos protocolos TCP/IP, que serão descritos na subseção 2.1.2.1, de ser uma comunidade

de pessoas que utilizam e desenvolvem essas redes e de ser uma coleção de recursos que podem ser acessados através dessas redes.

2.1.1 Origem

Para Kleinrock (2010), um dos pioneiros no desenvolvimento da tecnologia de redes de computadores, não é possível determinar um momento específico que deu origem à Internet. Segundo o autor, pode-se dizer que suas raízes estão nas tecnologias de comunicação mais antigas dos séculos e milênios passados, ou nos primórdios da matemática e da lógica, ou até mesmo com o surgimento da própria linguagem.

No entanto, é considerado que o marco inicial da revolução das comunicações tenha acontecido há cerca de 60 anos, no contexto da Guerra Fria. O desenvolvimento do projeto *Advanced Research Projects Agency Network* (ARPANET) pela Agência de Projetos de Pesquisa Avançada (ARPA) adotou e aperfeiçoou o protocolo TCP/IP, base fundamental para as comunicações da rede até os dias atuais. Também introduziu o conceito de comutação de pacotes, dividindo os dados em pacotes pequenos para transmissão.

Na década de 1960, a situação tecnológica do mundo era bem distinta da atual. Conforme o cientista Roberts (1988), responsável pela iniciação, planejamento e desenvolvimento do ARPANET, em 1964 só existiam computadores projetados para processar um grande volume de dados. Cada computador possuía seu próprio conjunto de usuários e o acesso simultâneo era abstruso, visto que seus terminais ficavam conectados fisicamente através de cabos ou por uma linha telefônica local. Dessa maneira, a troca de informações entre os computadores era feita através de fitas e não era possível acessar *softwares* de um outro computador.

Como mencionado anteriormente, a Guerra Fria estava em pleno andamento. Os Estados Unidos enfrentavam a necessidade de estabelecer uma comunicação robusta e confiável em caso de um possível ataque nuclear. Foi nesse contexto, e após o lançamento do Sputnik em 1957, pela União Soviética, que o Departamento de Defesa dos EUA estabeleceu a ARPA (HAUBEN, 2007), com o objetivo de unir pesquisadores e departamentos de pesquisa para que pudessem compartilhar

informações.

Em 1966, convencido de que redes de computadores eram importantes, Roberts realizou o primeiro experimento com conexão de computadores. Ele estabeleceu uma conexão de 1,2 kb/s, que era considerada alta velocidade na época, usando uma linha telefônica discada entre os computadores TX-2, do Laboratório de Lincoln em Lexington, Massachusetts e o computador Q-32, do System Development Corporation (SDC) em Santa Mônica, California (MARILL; ROBERTS, 1966). O objetivo deste experimento era explorar as capacidades de ambos os sistemas e promover a colaboração entre eles, permitindo que os dois computadores compartilhassem dados e recursos, o que era uma ideia inovadora.

As duas máquinas eram capazes de executar programas em tempo real. O TX-2 foi um dos primeiros computadores a usar transistores em vez de tubos de vácuo e a ter um monitor de vídeo (SUTHERLAND, 2012). Enquanto o Q-32 não limitava o número de usuários simultâneos nem aquisição de recursos por parte deles (HEMMENDINGER, 2014). Ao conectar os dois sistemas, Roberts e sua equipe puderam aproveitar o poder combinado dos computadores para realizar pesquisas e experimentos mais complexos.

O experimento foi um sucesso e foi um passo significativo no desenvolvimento de redes de computadores e também ajudou no desenvolvimento de técnicas de compartilhamento de recursos, como o acesso remoto a arquivos e o compartilhamento de dados entre computadores diferentes. Foi revelado que não houve problema em fazer com que os computadores se comunicassem ou usassem recursos um do outro, mas sim que a comunicação discada baseada na rede telefônica era muito devagar e não confiável. (ROBERTS, 1988)

A oportunidade de desenvolver um novo projeto de rede de computadores baseado em uma tecnologia completamente nova surgiu meses depois quando Roberts assumiu o comando dos projetos do IPT na ARPA enquanto ela patrocinava pesquisa nas principais universidades nos EUA. Com o auxílio dos pesquisadores dessas universidades, a ARPANET foi planejada durante 1967 com o objetivo de estabelecer uma comunicação entre esses computadores. A primeira tarefa foi desenvolver uma proto-

colo de interface que fosse compatível com todos os grupos de pesquisa. A segunda foi desenvolver uma nova tecnologia de comunicação que suportasse 35 computares em 16 localizações diferentes com um tráfego de dados de 500.000 pacotes por dia (ROBERTS, 1988).

Três anos após o experimento de Roberts, o professor Kleinrock e sua equipe realizaram o envio da primeira mensagem *host-to-host* através da ARPANET. A conexão foi realizada entre os computadores da Universidade da Califórnia, Los Angeles (UCLA) e o do Instituto de Pesquisa de Stanford (SRI), em Menlo Park, Califórnia, utilizando o Processador de Mensagens de Interface (IMP), que é um dispositivo projetado para rotear e interconectar redes de computadores, e foi estabelecida através de uma linha telefônica de 50 kb/s.

Uma vez estabelecida a conexão física entre os IMPs da UCLA e do SRI, um programa chamado Network Control Program (NCP), precursor do protocolo TCP/IP, foi utilizado para permitir a conexão de dados entre as duas máquinas. Dessa forma, a primeira mensagem transmitida através de uma conexão ocorreu quando a equipe de Kleinrock, enviou o comando *'login'* para o computador no SRI. (KLEINROCK, 2010).

Ainda segundo o professor, as duas equipes mantiveram contato por telefone enquanto a mensagem era enviada. A equipe do SRI confirmou o recebimento das letras 'l' e 'o' com sucesso, mas o sistema travou e o restante não foi entregue. Apesar deste contratempo, essa primeira conexão marcou o inicio do desenvolvimento das redes de computadores e estabeleceu as bases para a Internet moderna.

2.1.2 Protocolos da Internet

Para permitir a comunicação entre os computadores conectados à ARPANET, foi necessário desenvolver protocolos, que são conjuntos de regras e procedimentos que regulam a troca de informações em uma rede. O TCP/IP foi o conjunto de protocolos utilizado na ARPANET e tornou-se a base da Internet até os dias de hoje.

Com a sua implantação e a popularização da Internet, abordado na próxima seção,

surgiu a necessidade da criação de outros protocolos que atendessem às necessidades específicas que foram surgindo na comunicação na rede. Nesse sentido, destacam-se os protocolos SMTP, DNS e HTTP.

2.1.2.1 Arquitetura TCP/IP

Como definem Seth e Venkatesulu (2008), o Transmission Control Protocol (TCP) é um protocolo de comunicação orientado à conexão onde as extremidades precisam se comunicar para que compreendam os problemas uma da outra. Qualquer escassez de recursos em termos de memória ou CPU em uma ponta é comunicada à outra para que se tome medidas corretivas afim de diminuir a taxa de transação de dados. Isso evita a duplicação de esforços e tráfego desnecessário na rede.

O TCP é considerado um protocolo confiável porque ele registra cada *byte* de dados enviado e recebido pela outra extremidade, o que significa que qualquer perda de dados é detectada e tratada com cuidado. Além do mais, o TCP também é responsável por determinar a melhor maneira de enviar qualquer duplicação de esforços devido à perda de dados.

Trabalhando em conjunto com a camada de rede, o protocolo TCP consegue identificar a situação do tráfego de rede e, dependendo das condições, toma uma decisão sobre enviar dados em blocos menores ou maiores. Isso é conhecido como mecanismo de controle de congestionamento.

Já o Internet Protocool (IP) é responsável pelo roteamento dos pacotes de dados entre os dispositivos na rede. Ele fornece o endereçamento único para cada dispositivo conectado à Internet e determina a melhor rota para encaminhar os pacotes de dados de um ponto a outro. Socolofsky e Kale (1991) explicam que esse protocolo é fundamental para o sucesso da tecnologia da Internet. O cabeçalho IP contém o endereço IP, o que cria uma única rede lógica com origem em múltiplas redes físicas. Essa interconexão de redes físicas é a origem do nome "internet".

2.1.2.2 Protocolo de Correio Simples (SMTP)

Conforme supradito, novos protocolos foram desenvolvidos de acordo com as novas necessidades que surgiram ao longo da evolução da Internet. A comunicação por *e-mails* se difundiu rapidamente e, com isso, o protocolo SMTP foi desenvolvido com o objetivo de transferir o correio eletrônico de forma eficiente e confiável. (POSTEL, 1982)

O SMTP utiliza a abordagem cliente-servidor, em que um cliente de *e-mail* se conecta a um servidor de *e-mail* para enviar mensagens. O servidor de *e-mail* receptor, por sua vez, utiliza o protocolo para receber e armazenar os e-mails nas caixas de correio dos usuários. Uma de suas vantagens é a interoperabilidade. Ele permite que diferentes sistemas de e-mail se comuniquem de forma eficiente, independentemente do software ou do provedor de e-mail utilizado.

Conforme explica Riabov (2005), o SMTP é um protocolo de camada de aplicação usado como um mecanismo comum para o transporte de correspondência eletrônica entre diferentes hóspedes pelo TCP/IP. Sob o protocolo SMTP, um processo cliente abre uma conexão TCP com um processo servidor em um *host* remoto e tenta enviar um e-mail através dessa conexão. O servidor SMTP aguarda pela conexão TCP em uma determinada porta enquanto o processo cliente SMTP inicia uma conexão nessa porta. Após o sucesso da conexão TCP, os dois processos executam um diálogo simples de requisição e resposta no qual o processo cliente transmite os endereço do remetente e do destinatário, ou dos destinatários.

Quando o processo servidor aceita esses endereços de e-mail, o processo cliente transmite a mensagem de e-mail instantâneo, que deve conter um cabeçalho de mensagem e o texto da mensagem, chamado também de *corpo da mensagem*, formatado de acordo com o RFC 822 [1]. Os e-mails que chegam via SMTP são encaminhados para um servidor remoto ou são entregues às caixas de correio no servidor local. (RIABOV, 2005)

[1] https://www.rfc-editor.org/rfc/rfc822

2.1.2.3 Sistema de Nomes de Domínio (DNS)

Criado em 1983 por Paul Mockapetris e Jon Postel, o DNS foi a solução para o crescente problema de escalabilidade e complexidade em gerenciar a Internet em rápida ascensão. Mockapetris (1984) o descreve como um protocolo e um conjunto de servidores que fornece um método uniforme para associar os nomes dos recursos e.g., caixas de correio e nomes de *hosts*, às suas informações e.g, endereços de servidores de correio e endereços de rede.

O DNS é responsável por traduzir nomes de domínio em endereços IP. Ou seja, ao invés de memorizar sequências numéricas complexas, como um endereço de IP, por exemplo 192.168.0.1, é possível usar nomes de domínio entendíveis tal como exemplo.com.

Postel (1994) explica que existe uma hierarquia na nomenclatura dos domínios, onde existem o que são chamados de domínios de nível superior (TLD), para identificar diferentes tipos de organizações e fins. Os TDLs mais comuns são:

.com: Esse domínio foi inicialmente destinado a empresas comerciais. No entanto, é o domínio genérico mais conhecido atualmente e é utilizado para diferentes tipos de sites e.g., Google[2], Facebook[3] e Globo[4]

.edu: Esse domínio foi criado para registrar universidades, escolas e instituições educacionais. Nele estão presentes, por exemplo, o Instituto Federal do Espirito Santo (IFES)[5] e a Universidade Estadual do Norte do Paraná (UENP)[6]

.net: Esse domínio foi inicialmente destinado a provedores e empresas relacionadas à infraestrutura da Internet. Porém, atualmente o .net é usado por diversos sites, como por exemplo, o site da Microsoft[7]

.org: Esse domínio foi criado para ser destinado a organizações sem fins lucrativos,

[2]<https://google.com>
[3]<https://facebook.com>
[4]<https://globo.com>
[5]<https://www.ifes.edu.br/>
[6]<https://uenp.edu.br/>
[7]<https://microsoft.net/>

como a Cruz Vermelha[8] e o Green Peace[9]

.int: Esse domínio é reservado para organizações e tratados internacionais, tal qual a Organização Mundial de Saúde[10] .

.gov: Esse domínio é destinado as entidades governamentais, sejam elas federais, estaduais ou municipais. O Governo do Estado do Rio de Janeiro[11] é um exemplo de entidade presente nesse domínio.

2.1.2.4 Protocolo de Transferência de Hipertexto (HTTP)

O Protocolo de Transferência de Hipertexto (HTTP) é um protocolo utilizado para comunicação entre clientes, geralmente navegadores web, e servidores. Ele define a forma como as mensagens são formatadas e transmitidas e estabelece as regras para a transferência de dados entre o cliente e o servidor.

Berners-Lee, Fielding e Frystyk (1996) definem o protocolo HTTP como um protocolo de nível de aplicação para sistemas de informação distribuídos, colaborativos e de hipermídia. É um protocolo genérico e sem estado que pode ser usado para muitas tarefas além do uso para hipertexto, como servidores de nomes e sistemas de gerenciamento de objetos distribuídos, por meio da extensão de seus métodos de solicitação, códigos de erro e cabeçalhos. Uma característica do HTTP é a tipagem e negociação da representação de dados, permitindo que sistemas sejam construídos independentemente dos dados sendo transferidos.

Assim como o protocolo SMTP, também é um protocolo baseado na arquitetura cliente-servidor, onde o cliente envia requisições para o servidor e o servidor, por sua parte, responde com as informações solicitadas. Essas solicitações e respostas são realizadas através de mensagens compostas por um cabeçalho e, opcionalmente, um corpo de dados. Os métodos de requisição são definidos como a seguir:

OPTIONS: Usado para obter informações sobre as opções e recursos disponíveis

[8] <http://www.cruzvermelha.org.br/>
[9] <https://www.greenpeace.org/>
[10] <https://www.who.int/>
[11] <https://www.rj.gov.br/>

em um servidor.

HEAD: Usado para solicitar apenas os cabeçalhos de uma resposta HTTP, sem o corpo do recurso. É frequentemente usado para obter informações sobre um recurso sem transferir todos os dados.

POST: Usado para enviar dados ao servidor para a criação de um novo recurso.

PUT: Usado para enviar dados ao servidor e atualizar um recurso existente.

DELETE: Usado para solicitar a exclusão de um recurso específico.

TRACE: Usada para fins de depuração e diagnósticos. A resposta à uma requisição TRACE é exatamente a solicitação recebida.

CONNECT: Usado para estabelecer uma conexão de túnel bidirecional entre o cliente e o servidor por um *proxy*.

Além dessa definição, o protocolo HTTP classifica os métodos de requisições como idempotentes ou seguros. Métodos seguros são aqueles que não alteram o estado do servidor, são requisições realizadas apenas para obter informações ou recursos. Isso significa que fazer uma solicitação segura não deve ter efeitos colaterais no servidor.

Enquanto um método idempotente é aquele em que várias solicitações idênticas podem ser feitas de forma segura, sem causar efeitos colaterais adicionais além do efeito produzido pela primeira solicitação. Ou seja, a solicitação pode ser repetida várias vezes sem alterar o estado do servidor além da primeira solicitação. Os métodos GET, HEAD, PUT e DELETE compartilham essa propriedade. (FIELDING et al., 1999)

Além da definição dos métodos de requisição, o protocolo HTTP também define os códigos de status da resposta, que são categorizados de acordo com o primeiro dígito da seguinte forma:

1xx: Indica que a requisição foi recebida e está em processamento.

2xx: Indica que a solicitação foi recebida, entendida e processada com sucesso.

3xx: Indica que é necessária outra ação para que a requisição seja concluída.

4xx: Indica que houve algum erro na requisição feita pelo cliente, como uma URL inválida ou acesso negado.

5xx: Indica que ocorreu um erro no servidor durante o processamento da solicitação, como um erro interno ou tempo limite atingido.

2.1.3 Evolução

Por ser uma abordagem radicalmente diferente dos circuitos dedicados utilizados até então, Roberts (1988) conta que desde a primeira vez em que divulgou uma descrição de comutação de pacotes fora da comunidade de pesquisa em computação, que foi em 1967, até aproximadamente 1975, a reação da indústria das comunicações foi geralmente negativa. O defeito mais comum apontado pelos profissionais da comunicação, antes da construção da ARPANET, era que os *buffers* se esgotariam rapidamente. Após o funcionamento bem-sucedido da ARPANET, o discurso deles mudou e passaram a dizer que a comutação de pacotes nunca seria viável economicamente e que seria dependente do subsidio do governo.

Apesar de ter sido projetada para interligar os quatro nós principais da UCLA, do SRI, da Universidade da Califórnia em Santa Bárbara (UCSB) e a Universidade de Utah, a ARPANET foi sendo expandida e mais instituições de pesquisa, laboratórios e universidades foram sendo conectados, formando uma rede de comunicação cada vez mais ampla (LUKASIK, 2010). Embora fosse uma rede funcional, o acesso e a navegação pelos computadores e serviços disponíveis eram limitados aos especialistas em computação e pesquisadores. A interação com a rede era baseada em comandos de texto e não era intuitivo para os usuários navegarem pelos recursos disponíveis.

Em 1989, Tim Berners-Lee criou a WorldWideWeb (WWW), ou simplesmente Web, um sistema de informações que permitiu a visualização e acesso à vasta quantidade de conteúdo presente na Internet. Conforme os cientistas Berners-Lee et al. (1994), a Web foi desenvolvida com o objetivo de ser uma base de conhecimento, onde colaboradores ao redor do planeta pudessem compartilhar suas ideias. Ela

consiste em toda a estrutura que permite que os documentos e recursos sejam interligados e acessados através da Internet. Um ano mais tarde foi desenvolvido o primeiro *software* capaz de renderizar e exibir as páginas web. Tim Berners-Lee criou o navegador WorldWideWeb, ilustrado na Figura 2.1, que posteriormente foi renomeado para 'Nexus', afim evitar confusões com o servidor WWW.

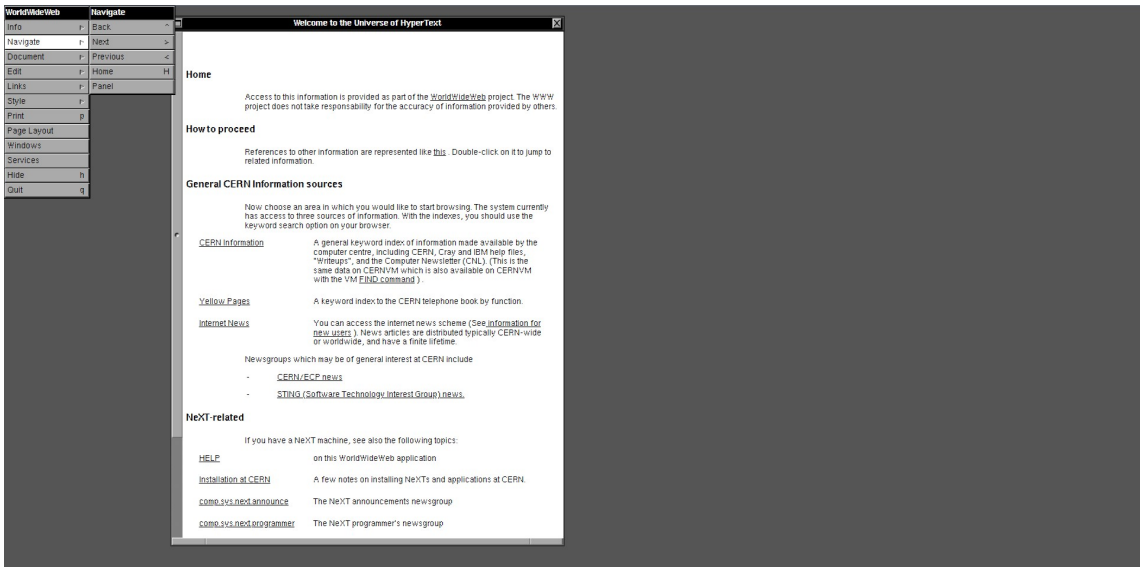

Figura 2.1: Página inicial do navegador WorldWideWeb

Entretanto, naquela época o navegador era executado apenas no computador NeXT, que, por sua vez, não era compatível com outros sistemas (BERNERS-LEE, 2004). Como a Web democratizou e revolucionou a forma como as pessoas passaram a acessar e compartilhar informações, tornou-se necessário que os navegadores fossem compatíveis com diferentes máquinas.

Dessa maneira e ainda de acordo com Berners-Lee (2004), em 1992 um grupo de estudantes na Finlândia desenvolveu dois navegadores compatíveis com o sistema operacional Windows[12]. E em 1993, o desenvolvimento do navegador Mosaic, pela equipe de Mark Andreeessen, do Centro Nacional para Aplicações de Supercomputação, contribuiu significativamente para o crescimento da Web.

Inspirado no sucesso Mosaic, em 1994 a Netscape Communications Corporation[13] desenvolveu o Netscape Navigator, que rapidamente se tornou bastante popular,

[12]<https://www.microsoft.com/pt-br/windows/>
[13]<https://www.britannica.com/topic/Netscape-Communications-Corp>

atingindo mais de 10 milhões de *downloads* apenas dois anos após seu lançamento (SEBENIUS, 2002). Em contrapartida, a Microsoft lançou o Internet Explorer[14] integrado com o sistema operacional Windows 95, o que deu inicio à chamada 'Guerra dos Navegadores', marcando a disputa pela predominância no mercado de navegadores entre as empresas citadas. Onde cada uma adotava uma versão diferente de alguma tecnologia inovadora e importante (PHILLIPS, 1998).

Em paralelo à competição pela liderança do mercado de navegadores entre Microsoft e Netscape, em 1994 Tim Bernes-Lee criou a organização internacional *World Wide Web Consortium* (W3C) [15], responsável pela consistência da Web, pela interoperabilidade entre navegadores e pela criação e manutenção de padrões como o HTML, CSS e Javascript. Ao longo dos anos, novos navegadores foram desenvolvidos e consolidados. Amplamente utilizados até hoje, o Safari[16], foi lançado em 2003 pela Apple[17], em 2004 o Firefox[18] foi lançado pela Mozilla Foundation[19], em 2008 a Google[20] desenvolveu seu próprio navegador, o Google Chrome[21]e em 2015 a Microsoft lançou o Microsoft Edge[22].

Antes da Web, o acesso à informação estava limitado aos sistemas de armazenamento e recuperação de dados, como bancos de dados ou bibliotecas físicas. Como foi dito anteriormente, a Web democratizou o acesso e compartilhamento de informação. Essa democratização contribuiu para sua evolução acelerada e, consequentemente, para o aumento significativo na quantidade de sites e de ferramentas e recursos para desenvolvê-los. Na próxima seção discursaremos brevemente sobre as principais tecnologias para o desenvolvimento Web.

[14] <https://learn.microsoft.com/pt-br/internet-explorer/internet-explorer>
[15] <https://www.w3.org/>
[16] <https://www.apple.com/br/safari/>
[17] <https://www.apple.com/br/>
[18] <https://www.mozilla.org/pt-BR/firefox/new/>
[19] <https://foundation.mozilla.org/pt-BR/>
[20] <https://about.google/intl/pt-BR/>
[21] <https://www.google.com/intl/pt-BR/chrome/>
[22] <https://www.microsoft.com/pt-br/edge/>

2.2 Desenvolvimento para Web

Para desenvolver uma aplicação para web é fundamental ter um bom entendimento sobre o funcionamento das linguagens de programação voltadas para web, responsáveis por construir e executar a lógica por trás do sistema. Nesta seção, discutiremos sobre algumas das linguagens web mais populares atualmente: *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript.

2.2.1 HTML

O HTML é uma linguagem de marcação utilizada para construir páginas da web. Nessa linguagem, os elementos da página são identificados com marcadores conhecidos como *tags*, e cada elementos desempenha uma função que contribui para a construção do *layout* da página. Esses elementos são interpretados e renderizados na página pelo navegador.

Dentro das *tags*, é possível adicionar atributos capazes de controlar o comportamento dos elementos. Existem dois tipos de atributos: os globais e os específicos. Os atributos globais são aceitos por todas as *tags*, como id e *style*. Já os específicos são aceitos somente por algumas *tags*, como *href* e *disabled*.

Na figura 2.2 é possível visualizar à esquerda o código HTML e, à direita, o resultado após renderização do navegador. No código estão algumas das *tags* mais importantes do HTML. São elas:

1. ***<html>*** Define o início e o fim de um documento HTML.
2. ***<head>*** Contém informações de metadados sobre o documento.
3. ***<meta charset="UTF-8"/>*** Especifica que o documento está codificado com UTF-8 [23].
4. ***<title>*** Define o título da página.

[23] UTF-8 (*UCS Transformation Format 8*) é a codificação de caracteres que converte um Unicode ou um número hexadecimal em uma sequência específica de *bytes*. O padrão Unicode mapeia o código para um conjunto de caracteres com o objetivo de padronizar a representação computacional dos sistemas web em todo mundo.(KILBOURNE; WILLIAMS, 2003)

5. ***<body>*** Representa o corpo do documento.

6. ***<h1>*** Representam títulos que variam de tamanho de acordo com o número usado (h1 até h6).

7. ***<p>*** Representa um parágrafo.

8. ***<button>*** Representa um botão.

Figura 2.2: Exemplo de código de HTML e renderização

A versão mais atual do HTML é o HTML5. Essa versão oferece algumas novas funcionalidades como o HTML semântico, com *tags* que ajudam a estruturar o conteúdo da página de maneira clara e significativa, suporte nativo a áudio e vídeo, permitindo reprodução de mídia sem uso de *plugin*, e o novo elemento *<canvas>*, que permite desenhar componentes gráficos.

Utilizando a ampla variedade de *tags* disponíveis no HTML é possível construir páginas web estruturadas e funcionais. Entretanto, muitas vezes os elementos gerados não estão estilizados de acordo com o desejado. Nesses casos, para aprimorar a apresentação visual da página, é necessário utilizar, em conjunto com o HTML, o CSS.

2.2.2 CSS

O CSS é a linguagem utilizada para estilizar os elementos gerados pelo HTML. Ele foi desenvolvido em 1996 pelo W3C[24] por uma razão simples: resolver o problema gerado pela falta de *tags* para estilizar as páginas web. (DOMANTAS, 2023)

Dentre as possibilidades oferecidas pelo CSS estão: controlar atributos como tamanho, cor e posição dos elementos da página, controlar o *design* da página, tornando-o responsivo para se adaptar a telas de diversos tamanho, e criar animações e transições para adicionar efeitos visuais aos elementos.

É possível aplicar um estilo CSS diretamente a todos os elementos HTML de uma determinada *tag*, ou somente a elementos que possuam uma classe específica. Essa flexibilidade é fundamental para garantir que o estilo CSS seja aplicado apenas aos elementos desejados.

Na figura 2.3 podemos observar a *tag <style>* no código HTML. Dentro dessa *tag*, está o CSS responsável por estilizar os elementos na página. O CSS aplicou a propriedade *font-size: 25px* diretamente ao elemento *<p>*, resultando no aumento do tamanho da fonte de ambos os parágrafos. No entanto, somente o parágrafo com o atributo *class="p1"* recebeu o estilo aplicado a essa classe: a cor vermelha.

Além disso, o CSS aplicou a propriedade *padding: 10px* ao elemento *button*, o que resultou no aumento do tamanho do botão devido ao preenchimento de 10 *pixels* nas margens internas do elemento.

Embora o CSS seja uma linguagem poderosa, lidar com a criação estilos padronizados para grandes aplicações pode ser um desafio. É nesse contexto que surgem os *frameworks* de CSS, bibliotecas que oferecem classes de CSS com estilos pré-definidos. Dentre os *frameworks* mais populares e amplamente utilizados atualmente estão o Bootstrap[25], o TailwindCSS[26] e o Bulma[27].

A versão mais recente do CSS é o CSS3. Com essa versão, tornou-se possível

[24] <https://www.w3.org>
[25] <https://getbootstrap.com>
[26] <https://tailwindcss.com>
[27] <https://bulma.io>

Figura 2.3: Exemplo de código de HTML e CSS e renderização

aplicar estilos mais complexos e sofisticados como sombras, opacidade, animações e transições. Com o CSS3 também surgiram as *media queries*, que permitem aplicar estilização a elementos de acordo com condições específicas, como o tamanho da tela ou o tipo de dispositivo.

O CSS desempenha um papel fundamental na construção da interface de uma aplicação. No entanto, quando se busca adicionar interatividade e funcionalidades dinâmicas à aplicação, é necessário recorrer a uma linguagem de programação como o JavaScript.

2.2.3 JavaScript

O JavaScript é uma linguagem de *script* orientada a objeto, usada para desenvolver páginas interativas na web. De acordo com Kiran (2023), o JavaScript está presente em aproximadamente 98.7% dos *websites*, evidenciando sua popularidade. Com o JavaScript, é possível manipular elementos HTML, responder eventos do usuário, como cliques de botão, e atualizar dinamicamente conteúdos na página.

Dentre as vantagens de utilizar JavaScript estão: execução *client-side*, o que significa que o próprio navegador é responsável por processar o código, a compatibilidade com os navegadores modernos, e a possibilidade de dinamizar a página web. Além disso, o JavaScript oferece a capacidade de comunicação assíncrona com APIs, permitindo buscar e enviar dados sem a necessidade de recarregar a página.

O JavaScript possui um ecossistema robusto devido à ampla variedade de *frameworks* disponíveis, como Angular[28], React[29] e Vue.js[30], projetados para acelerar o desenvolvimento e promover uma arquitetura consistente, oferecendo componentes reutilizáveis, e compatibilidade com diversos navegadores e plataformas.

Em suma, HTML, CSS e Javascript formam um trio poderoso para desenvolver aplicações web. Essas tecnologias permitem construir *websites* visivelmente atraentes, responsivos e interativos, proporcionando experiências ricas ao usuário.

[28] <https://angular.io>
[29] <https://react.dev>
[30] <https://vuejs.org>

Capítulo 3

Sistema de Avaliação de Disciplina

A avaliação de disciplinas por parte dos alunos no ensino superior é uma ferramenta importante para fornecer *feedback* à instituição de ensino e aos professores sobre a qualidade do ensino e a experiência do aluno. Essa avaliação permite que os alunos expressem suas opiniões, avaliem a eficácia dos métodos de ensino e forneçam sugestões para melhorias sobre a qualidade dos materiais didáticos, a clareza das instruções, a relevância do conteúdo apresentado, a organização do curso, a disponibilidade e conduta do professor.

Essa prática é valiosa, visto que os alunos estão diretamente envolvidos no processo de aprendizado e têm uma perspectiva única sobre o impacto das estratégias de ensino em seu desenvolvimento acadêmico. Além disso, a avaliação dos alunos pode fornecer informações adicionais sobre aspectos não observados pelos professores, como a dinâmica da sala de aula, a interação entre os estudantes e o ambiente de aprendizagem.

Este capítulo, na seção 3.1, expõe a problemática acerca da avaliação do ensino superior sendo exercida de forma genérica e sugere a necessidade de uma avaliação interna e que seja realizada de forma automatizada. Já a seção 3.2 é responsável pela apresentação de projetos web, semelhantes à este proposto, acerca da avaliação institucional do ponto de vista do discente onde apresentamos algumas universidades que realizam essa prática. Enquanto cabe à seção 3.3 apresentar a modelagem

do projeto apresentado, contando com a identificação dos atores e a descrição das funcionalidade do sistema.

3.1 Motivação

Por meio da Lei n 10.861 (BRASIL, 2004), as instituições de ensino superior no país devem estar de acordo com as normas do Sistema Nacional de Avaliação da Educação Superior (SINAES) cujo objetivo é contribuir para a melhoria contínua dos cursos e universidades avaliando os aspectos relacionados ao ensino, pesquisa, extensão, corpo docente, gestão institucional, bem como a responsabilidade social e infraestrutura.

A avaliação institucional desempenha um papel crucial na melhoria contínua da qualidade educacional e é uma tarefa complexa que demanda a consideração de vários fatores. Como apontam Polidori, Marinho-Araujo e Barreyro (2006), os desafios do SINAES são muitos e dependem de uma implementação fiel à proposta original favorecendo e fomentando o desenvolvimento dos processos formativos decorrentes da auto-avaliação. As informações e análises qualitativas nos três pilares vêm a beneficiar a difusão de uma cultura da avaliação que não se resume à construção de uma simples lista com o ranking de instituições. Ou seja, embora o SINAES seja fundamental para esse processo no Brasil, sua metodologia geral de avaliação é aplicada a todas as esferas do conhecimento, o que pode não levar em conta especificidades de determinadas áreas.

Para que a instituição esteja preparada para enfrentar desafios, é necessário que seus membros tenham ciência de sua realidade, virtudes, capacidades e limitações. De acordo com Júnior (2009), os processos avaliativos precisam envolver o maior número de participantes, tanto na construção de seu projeto quanto na análise e no uso dos resultados, contribuindo para o desenvolvimento humano.

Os discentes são os principais protagonistas desse processo, sendo diretamente impactados pelas disciplinas cursadas e pelos professores que as ministraram. Ao avaliar esses elementos, os estudantes podem manifestar suas opiniões, pontuar se o

conteúdo está alinhado às expectativas e se é relevante para sua formação. Além disso, uma avaliação do professor possibilita uma análise aprofundada do seu desempenho, levando em conta didática, comunicação e capacidade de estimular o interesse do aluno.

Nesse sentido, e como afirmaram Mohammadi, Eshaghi e Arefi (2012), as instituições devem se esforçar para estabelecer um sistema interno de garantia da qualidade que forneça uma base genuína para a tomada de decisões de gestão. Levando em consideração o contexto no qual vivemos, onde a tecnologia influencia para além das atividades cotidianas, torna-se essencial a utilização de um sistema como este proposto, que permite às universidades coletarem, processarem e analisarem os dados de maneira precisa e eficiente, respeitando a privacidade dos alunos.

A automação desse processo traz uma vantagem significativa ao reduzir substancialmente o tempo necessário para a obtenção dos resultados. Com a utilização de sistemas automatizados, a coleta, processamento e análise dos dados acontecem de forma eficiente e ligeira, resultando em informações pertinentes para a instituição. Isso permite que as universidades obtenham dados quantitativos e qualitativos abrangentes de maneira mais rápida, otimizando assim a tomada de decisões e possibilitando uma análise precisa do desempenho das disciplinas e dos docentes.

3.2 Trabalhos Relacionados

De uma maneira mais geral, existe um *website* bastante popular nos Estados Unidos chamado *Rate my Professors*[1], que permite que estudantes pesquisem e avaliem professores e universidades de todo o país. Esse recurso foi desenvolvido com o objetivo de ajudar estudantes universitários a escolherem suas aulas e professores de forma mais informada, tendo como base as opiniões de outros estudantes. O *site* fornece *feedback* dos usuários sobre a metodologia de ensino dos professores e suas disciplinas, incluindo avaliações das instalações dos campi universitários, como mostra a figura 3.1.

[1] <https://www.ratemyprofessors.com/>

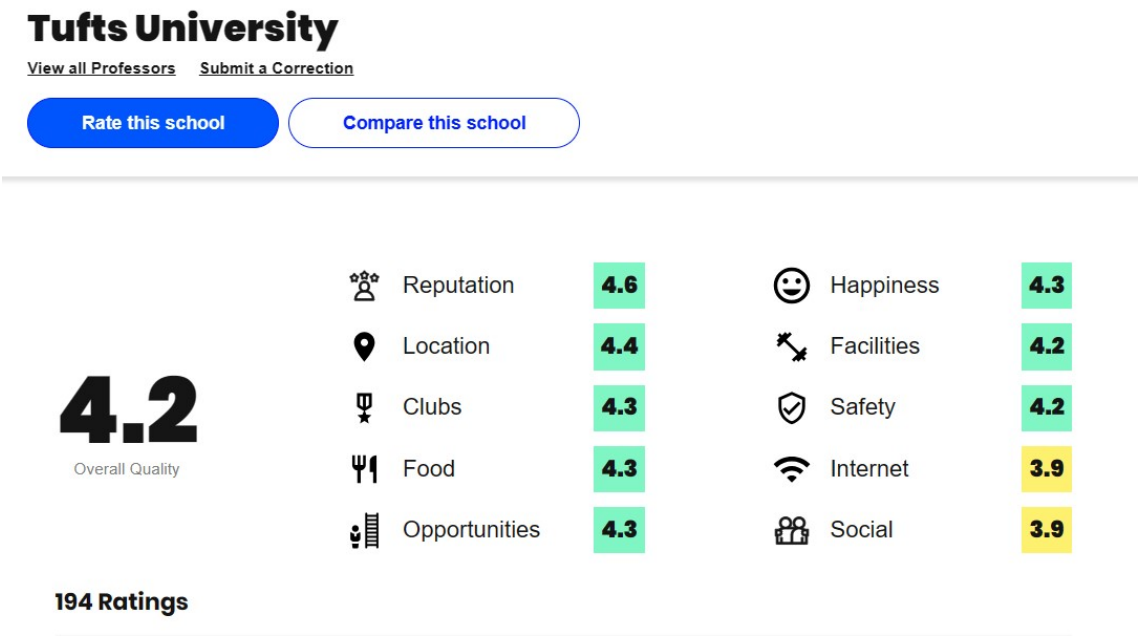

Figura 3.1: Avaliação da Tufts University disponível no site ratemyprofessors

Fonte: Rate My Professors

Retratando casos de universidade em diferentes contextos que adotam a auto-avaliação como parte do seu sistema interno de avaliação, a *Universidad de Chile*, por exemplo, conta com a Comissão Superior de Autoavaliação Institucional (CSAI), um órgão colegiado responsável por orientar, administrar e analisar os processos de avaliação interna. Conforme o próprio *Regulamento de Autoavaliação Institucional*[2] da universidade, seus objetivos são desenvolver e fortalecer os mecanismos de autor-regulação da qualidade e coletar e e sistematizar informações sobre o funcionamento e progresso da Instituição, garantindo a existência de processos eficazes.

No Brasil, a USP conta com um sistema de avaliação de disciplinas interno que, semelhantemente ao *Rate my Professors*, busca ajudar novos alunos a tomarem decisões acerca das matérias e professores baseando-se em avaliações feitas pela comunidade estudantil. Através do USP Avalia[3], os estudantes têm a oportunidade de avaliar cada disciplina cursada no período letivo, incluindo aspectos como qualidade do professor, conteúdo programático e métodos de ensino. As avaliações ficam

[2] <https://uchile.cl/presentacion/prorrectoria/proceso-de-autoevaluacion-institucional/normativa-vinculada/reglamento-de-autoevaluacion-institucional>

[3] <https://uspavalia.com/>

disponíveis para serem consultadas no sistema, onde é exibida a média de avaliação da disciplina conforme exibido na imagem 3.2 a seguir:

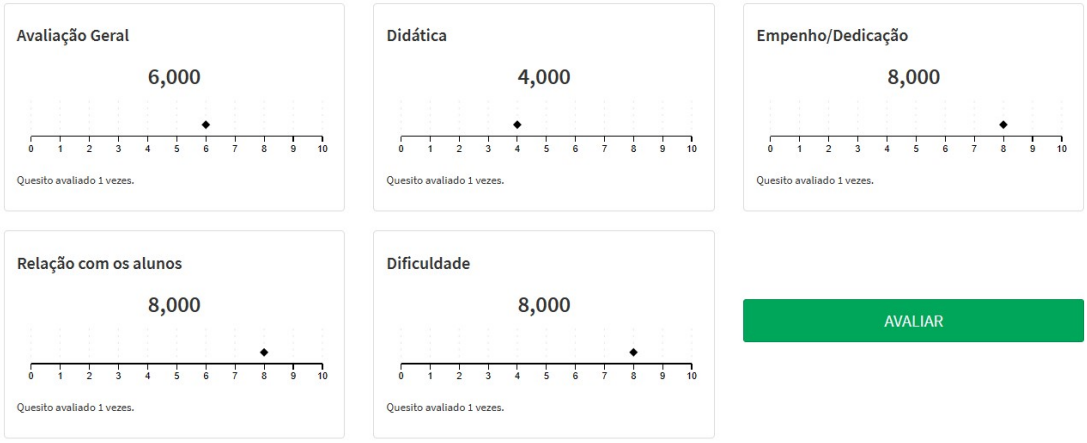

Figura 3.2: Avaliação da disciplina SCC0260 - Interação Usuário-Computador da USP

Fonte: USP Avalia

Já no Rio de Janeiro, Universidade Federal Fluminense (UFF) possui a Comissão Própria de Avaliação Institucional da UFF (CPA/UFF), responsável pela criação do Sistema de Avaliação Institucional (SAI)[4], um instrumento desenvolvido para coletar a perspectiva dos alunos e professores sobre os cursos de graduação, mediante a avaliação do desempenho efetivo nas disciplinas e da disponibilidade de infraestrutura para o funcionamento dos cursos. O SAI é bem robusto e nele é possível gerar relatórios das avaliações de discplinias realizadas pelos discentes separados por departamento, por curso, por área e até mesmo por localidade. A seguir, na imagem 3.2, consta o relatório referente à avaliação da disciplina Programação de Computadores I (RCM00002), do Departamento de Computação, relativo ao ao segundo período letivo do ano de 2021. Os resultados das avaliações são analisados pela CPA/UFF, pelas Unidades Acadêmicas, pelos Departamentos de Ensino e pelas Coordenações de Curso e fornecem uma base importante para refletir sobre a qualidade do trabalho acadêmico realizado na UFF, além de gerar informações relevantes e necessárias para a reformulação dos Projetos Pedagógicos dos Cursos.

[4]<https://app.uff.br/sai>

Pergunta	Não se aplica / Não sei avaliar	Discordo Totalmente	Discordo Parcialmente	Concordo Parcialmente	Concordo Totalmente
O(a) professor(a) utilizou técnicas de ensino e recursos didáticos inovadores e que contribuíram para a minha aprendizagem	1 (4.0 %)	0 (0.0 %)	2 (8.0 %)	8 (32.0 %)	14 (56.0 %)
As avaliações de aprendizagem foram coerentes com o nível de profundidade dos temas trabalhados pelo(a) professor(a)	1 (4.0 %)	1 (4.0 %)	3 (12.0 %)	1 (4.0 %)	19 (76.0 %)
O(a) professor(a) utilizou técnicas de ensino e avaliações de aprendizagem acessíveis aos(às) estudantes com deficiência, quando necessário	7 (28.0 %)	1 (4.0 %)	1 (4.0 %)	2 (8.0 %)	14 (56.0 %)
O(a) professor(a) utilizou técnicas de ensino e avaliações de aprendizagem acessíveis aos(às) estudantes com dificuldades de acesso digital, quando necessário	5 (20.0 %)	1 (4.0 %)	4 (16.0 %)	2 (8.0 %)	13 (52.0 %)
A bibliografia indicada contribuiu para meu estudo e aprendizagem	3 (12.0 %)	1 (4.0 %)	3 (12.0 %)	2 (8.0 %)	16 (64.0 %)
O(a) professor(a) me incentivou a refletir sobre os temas da disciplina	3 (12.0 %)	1 (4.0 %)	1 (4.0 %)	8 (32.0 %)	12 (48.0 %)
O(a) professor(a) ofereceu oportunidades para o esclarecimento de dúvidas	1 (4.0 %)	2 (8.0 %)	2 (8.0 %)	4 (16.0 %)	16 (64.0 %)
Houve diálogo na relação professor(a)-aluno(a)	1 (4.0 %)	3 (12.0 %)	5 (20.0 %)	1 (4.0 %)	15 (60.0 %)
O professor(a) cumpriu o horário das aulas na forma prevista no plano de atividades da disciplina	1 (4.0 %)	1 (4.0 %)	1 (4.0 %)	2 (8.0 %)	20 (80.0 %)
O(a) professor(a) propôs atividades assíncronas compatíveis com a carga horária da disciplina	3 (12.0 %)	1 (4.0 %)	1 (4.0 %)	2 (8.0 %)	18 (72.0 %)
O(a) professor(a) demonstrou dominar o conteúdo da disciplina	1 (4.0 %)	1 (4.0 %)	1 (4.0 %)	3 (12.0 %)	19 (76.0 %)
O(a) professor(a) cumpriu o plano de aula proposto, adequando-o às especificidades da turma	1 (4.0 %)	1 (4.0 %)	1 (4.0 %)	3 (12.0 %)	19 (76.0 %)
Total	**28 (9.3 %)**	**14 (4.7 %)**	**25 (8.3 %)**	**38 (12.7 %)**	**195 (65.0 %)**

Figura 3.3: Relatório de avaliação da disciplina RCM00002 - Programação de Computadores I, do Departamento de Computação (RCM), referente ao período 2021/2.

Fonte: SAI

3.3 Proposta

Conforme supracitado, a qualidade do ensino é um elemento crucial para o desenvolvimento dos estudantes e para o aprimoramento das instituições de ensino. Visando promover uma educação cada vez mais eficiente e participativa, este trabalho propõe o desenvolvimento de um sistema *web* de avaliação de disciplinas para o curso de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro, com o objetivo de oferecer *feedback* periódico e valioso para o aperfeiçoamento contínuo do processo educacional e viabilizando uma relação de transparência e qualidade no ambiente acadêmico.

O sistema tem como objetivo analisar o desempenho das disciplinas ofertadas no período letivo decorrido, coletando informações sobre a eficácia, organização e relevância. Os alunos serão estimulados a avaliar aspectos como o conteúdo ministrado, a metodologia aplicada, avaliações e recursos didáticos afim de identificar pontos fortes e fracos de cada uma bem como examinar a atuação dos professores, buscando avaliar o desempenhos dos docentes no processo de ensino-aprendizagem,

serão considerados critérios como o domínio sobre a matéria, didática, clareza sobre a ementa, comunicação compreensiva, disponibilidade e capacidade de manter os alunos motivados.

Para a realização da avaliação, foi desenvolvida uma plataforma web que permite aos alunos expressar suas opiniões de forma segura e anônima e aos professores e gestores acompanhar as métricas através de uma análise das avaliações. Os resultados serão apresentados de maneira estruturada e compreensível em um *dashboard*, que permitirá que os professores acompanhem suas próprias avaliações e que os coordenadores tenham acesso às avaliações do corpo docente sob o qual é responsável.

Foram elaborados questionários estruturados que abordam os diferentes aspectos relacionados às disciplinas cursadas e aos professores que as ministraram. Os questionários são compostos com questões de escalas de avaliação de 1 a 5, sendo 1 muito ruim e 5 muito bom, e campos de comentários abertos para a identificação de pontos positivos e negativos. É importante ressaltar a confidencialidade das respostas dos estudantes. O sistema garantirá o anonimato dos discentes, encorajando a participação franca e honesta permitindo que os estudantes sintam-se a vontade para expressar suas opiniões de forma livre e sem receio de represálias.

Os professores terão acesso a uma visão geral das avaliações recebidas em suas disciplinas. Eles conseguirão visualizar os resultados de forma consolidada, como médias e comentários feitos pelos discentes, podendo compreender com mais clareza o *feedback* recebido e identificar áreas de destaque e oportunidade de melhorias, além de admitir uma análise com avaliações anteriores e com as médias da instituição. Já para os coordenadores será disponibilizada uma análise mais abrangente, tendo acesso às avaliações dos professores sob sua responsabilidade, permitindo uma análise comparativa e um acompanhamento mais efetivo do desempenho docente.

Essa iniciativa busca fortalecer o diálogo entre alunos e instituição, disponibilizando avaliações de disciplinas e professores, bem como sua análise, para a comunidade acadêmica. A finalidade é aprimorar o gerenciamento das informações, promover transparência, autoaperfeiçoamento docente e demonstrar o compromisso da universidade em atender as necessidades e expectativas dos estudantes com base

em dados.

3.3.1 Atores

Um ator é uma entidade externa que interage com o sistema e desempenha um papel específico dentro do contexto do sistema em análise. Ao identificar os atores, é possível determinar quais são as suas responsabilidades e como eles se relacionam com as funcionalidades do sistema. Essas informações são utilizadas para modelar os casos de uso e requisitos do sistema.

Com base na proposta apresentada anteriormente, é possível identificar os seguintes atores:

1. **Aluno**: Representa os alunos do curso de Ciência da Computação. Eles podem visualizar a lista de disciplinas em que estão matriculados e avaliá-las uma vez, após o encerramento do período letivo.
2. **Professor**: Representa os professores responsáveis pelas disciplinas do curso. Eles podem visualizar as disciplinas que lecionam e acompanhar um relatório com as avaliações recebidas para cada uma delas.
3. **Coordenador**: Representa o(s) coordenador(es) do curso. Eles têm a responsabilidade de cadastrar disciplinas, associar os professores às disciplinas ministradas por eles, e, principalmente, visualizar relatórios com as avaliações de cada disciplina. Esses relatórios são fundamentais para análise de resultados e tomadas de decisões estratégicas para melhorar a qualidade do curso.

Esses atores desempenham papéis distintos no sistema de avaliação de disciplinas, e cada um contribui de uma maneira para o funcionamento do sistema.

3.3.2 Questionário

O questionário de avaliação de disciplina e do docente foi elaborado para capturar informações pertinentes acerca da experiência dos estudantes em relação à disciplina

cursada, permitindo a identificação de pontos fortes e áreas passíveis de aprimoramento. Ele conta com sete questões avaliativas de escala, de 1 a 5, que abordam aspectos como didática, pontualidade, assiduidade, capacidade de cumprir a ementa, interdisciplinaridade e aplicação de avaliações, além de duas questões discursivas para que os alunos expressem suas opiniões de forma mais livre.

Avaliar tais aspectos é importante para garantir a qualidade do ensino superior. Eles são cruciais uma vez que a didática eficaz ajuda na compreensão dos alunos, pontualidade e assiduidade demonstram respeito e comprometimento do docente para com os estudantes e com todo o processo educativo; o cumprimento da ementa garante que o curso está sendo ministrado corretamente de acordo com as expectativas e com os requisitos estabelecidos. Já a interdisciplinaridade enriquece o aprendizado, as avaliações são importantes para verificar o progresso dos alunos, e o conteúdo cobrado precisa ser relevante para a formação do aluno e estar alinhado com os objetivos do curso.

Optamos pela utilização de um conjunto de questões pré definidas visando a padronização das respostas e, também, para assegurar que os objetivos da organização sejam atendidos de forma adequada. Outra vantagem é a comparabilidade dos resultados, sendo possível comparar diferentes disciplinas, professores e períodos letivos e auxiliar a administração acadêmica na tomada de decisões informadas e na implementação de ações corretivas efetivas.

Ademais, o uso de um conjunto de questões padronizadas proporciona uma experiência mais consistente para os estudantes. Isso lhes permite ter uma noção clara do que esperar e responder às perguntas de forma mais direcionada, além de possibilitar a comparação de suas experiências em diferentes disciplinas. Oferecendo, assim, uma perspectiva mais abrangente sobre a qualidade do ensino na universidade.

Dessa forma, a seguir apresentamos as sete questões utilizadas no sistema de avaliação de disciplina proposto:

1. Com base no período corrente quanto você avalia que foi a didática do(a) professor(a)?

2. Com base no período corrente quanto você avalia que foi a pontualidade do(a) professor(a)?

3. Com base no período corrente quanto você avalia que foi a assiduidade do(a) professor(a)?

4. Com base no período corrente quanto você avalia que o(a) professor(a) cumpriu a ementa da disciplina?

5. Você conseguiu perceber, com base nas aulas do(a) professor(a) a interdiciplinalidade do conteúdo dado?

6. Com base no período corrente quanto você avalia que foram as avaliações (trabalhos, provas, teste, etc) do(a) professor(a)?

7. Com base no período corrente quanto você avalia a forma de cobrar o conteúdo dado pelo(a) professor(a)?

E aqui estão as questões onde o aluno pode se expressar com suas palavras:

1. Faça uma breve descrição dos pontos positivos percebidos nessa disciplina

2. Faça uma breve descrição dos pontos negativos percebidos nessa disciplina

3.3.3 Regras de Negócio

As Regras de Negócio (RNs) definem diretrizes e restrições de uma aplicação. Elas são essenciais para que haja clareza do que deve ser feito ao desenvolver um sistema, atendendo às necessidades e requisitos do negócio. Algumas das regras de negócio aplicadas ao sistema de avaliação de disciplinas são:

1. **RN01 - Usuários autenticados podem interagir com o sistema**

   Ao acessar o sistema, todos os usuários devem informar suas credenciais de acesso, *e-mail* e senha. Somente após a autenticação bem-sucedida, eles terão permissão para interagir com as funcionalidades e recursos do sistema de acordo com as limitações de cada perfil.

2. **RN02 - Usuários podem se registrar no sistema**

   Caso o usuário não tenha permissão para acessar sistema, ele pode realizar seu cadastro informando seu nome, *e-mail*, matrícula ou siape, perfil (aluno ou professor), se é administrador e uma senha.

3. **RN03 - Usuários registrados como aluno podem avaliar disciplinas**

   Apenas usuários com perfil aluno no sistema tem acesso ao questionário de avaliação de disciplina. Caso o usuário possua um perfil diferente de aluno, essa funcionalidade não estará disponível.

4. **RN04 - Alunos só podem avaliar disciplinas que cursaram no período letivo**

   A avaliação de uma disciplina só é permitida para alunos que estiveram matriculados nela durante o período letivo. Se um aluno não estiver matriculado em uma disciplina específica, ele não terá permissão para realizar a avaliação dessa disciplina.

5. **RN05 - Alunos só podem avaliar disciplinas uma única vez no período letivo**

   Para cada período letivo, um aluno só pode realizar a avaliação de uma disciplina uma vez. Após a realização da avaliação de uma disciplina, o aluno não poderá repetir ou refazer a avaliação para a mesma disciplina no mesmo período letivo.

6. **RN06 - Confidencialidade das respostas dos alunos**

   O sistema deve garantir que a identidade do aluno não seja revelada no processo de avaliação.

7. **RN07 - Usuários registrados como professor podem visualizar suas análises**

   Usuário com perfil professor só pode visualizar as avaliações atribuídas a ele mesmo nas disciplinas que ministrou. As avaliações de outros professores em disciplinas diferentes não são acessíveis para visualização do professor.

8. **RN08 - Usuários registrados como coordenador podem visualizar relatórios gerais**

   Apenas usuários com o perfil coordenador no sistema tem permissão para visualizar os relatórios gerais das disciplinas e professores do departamento. Outros usuários não têm acesso a esses relatórios.

9. **RN09 - Usuários registrados como coordenador podem cadastrar disciplinas**

   Apenas o coordenador do departamento tem a capacidade de adicionar novas disciplinas ao sistema e realizar alterações em disciplinas existentes.

10. **RN10 - Usuários registrados como coordenador podem cadastrar disciplinas**

    Apenas o coordenador do departamento tem a capacidade de adicionar novas disciplinas ao sistema e realizar alterações em disciplinas existentes.

11. **RN11 - Usuários registrados como coordenador podem cadastrar períodos letivos**

    Apenas o coordenador do departamento tem a permissão de incluir um novo período letivo no sistema e realizar alterações em períodos existentes.

12. **RN12 - Serão ofertadas disciplinas do Departamento de Ciência da Computação**

    Somente as disciplinas pertencentes ao Departamento de Ciência da Computação serão disponibilizadas para avaliação pelos estudantes.

Esses exemplos destacam algumas das regras de negócio cruciais para o funcionamento do sistema de avaliação de disciplinas. Ao seguir essas regras, é possível desenvolver uma aplicação segura, eficiente e que respeite a privacidade dos dados dos usuários.

3.3.4 Requisitos do Sistema

Os requisitos do sistema são descrições sobre o que o sistema deve ser capaz de fazer para atender às regras de negócios. Esses requisitos são divididos em requisitos funcionais e não funcionais.

Os Requisitos Funcionais (RFs) especificam funcionalidades que o sistema deve ter para atender às necessidades do usuário final. Já os requisitos não funcionais especificam as restrições e limitações da aplicação.

Alguns requisitos funcionais do Sistema de Avaliação de Disciplina proposto são:

1. **RF01 - Autenticar usuário**

   O sistema deve permitir o acesso do usuário ao serem informados os dados *e-mail* e senha.

2. **RF02 - Cadastrar usuário**

   O sistema deve permitir o cadastro de usuário ao serem informados nome, *e-mail*, matricula ou siape, perfil (aluno ou professor), se é administrador e uma senha.

3. **RF03 - Cadastrar disciplina**

   O sistema deve permitir que o coordenador cadastre disciplinas, fornecendo nome, código, turno, período letivo e professor.

4. **RF04 - Editar disciplina**

   O sistema deve permitir que o coordenador edite as informações de uma disciplina cadastrada, podendo alterar o nome, código, turno, período letivo e professor.

5. **RF05 - Excluir disciplina**

   O sistema deve permitir que o coordenador exclua disciplinas previamente cadastradas, garantindo que todas as informações relacionadas sejam removidas do sistema de forma permanente.

6. **RF06 - Listar disciplina**

   O sistema deve permitir que as disciplinas cadastradas sejam exibidas para os usuários.

7. **RF07 - Avaliar disciplina**

   O sistema deve permitir que o aluno avalie as disciplinas cursadas e os professores que as ministraram.

8. **RF08 - Cadastrar período letivo**

   O sistema deve permitir que o coordenador insira períodos letivos no sistema, fornecendo nome e o atributo *default*, que indica se o período é o período atual.

9. **RF09 - Editar período letivo**

   O sistema deve permitir que o coordenador edite as informações de um período letivo, podendo alterar o nome.

10. **RF10 - Excluir período letivo**

    O sistema deve permitir que o coordenador exclua períodos registrados no sistema, garantindo que todas as informações relacionadas sejam removidas de forma segura.

11. **RF11 - Visualizar relatórios gerais**

    O sistema deve permitir que o coordenador acesse os relatórios das avaliações e métricas de todas as disciplinas e professores do Departamento de Ciência da Computação.

12. **RF12 - Visualizar avaliação individual**

    O sistema deve permitir que o professor acesse os resultados das avaliações realizadas pelos alunos e métricas das disciplinas que eles próprios ministram.

Ao satisfazer todos os requisitos do sistema, podemos garantir que o software é confiável e funcional.

3.3.5 Casos de Uso

Os Casos de Uso (UCs) representam interações entre os atores e o sistema. Eles documentam as principais funcionalidades do sistema do ponto de vista do usuário final.

Alguns casos de uso do sistema de avaliação de disciplina são:

1. **UC01 - Efetuar *login***

   Atores Aluno, Professor, Coordenador.

   Descrição Autenticação de usuários cadastrados no sistema, permitindo a realização de operações restritas à cada perfil.

   Fluxo básico

   (a) O usuário acessa o sistema.

   (b) O sistema exibe o formulário de autenticação, solicitando os dados obrigatórios *e-mail* e senha.

   (c) O usuário informa suas credenciais.

   (d) O sistema valida as credencias fornecidas.

   (e) Se os dados forem válidos, o usuário é autenticado e redirecionado para a página de listagem de disciplinas.

   (f) Caso contrário, é exibida uma mensagem informando o erro.

2. **UC02 - Cadastrar-se no sistema**

   Ator Usuário.

   Descrição Inclusão do registro do usuário Aluno, Professor ou Coordenador no sistema.

   Fluxo básico

   (a) O usuário acessa o sistema.

   (b) O sistema exibe o formulário de autenticação, solicitando *e-mail* e senha a opção re realizar cadastro.

(c) O usuário seleciona a opção de se cadastrar no sistema.

(d) O sistema exibe o formulário de cadastro, solicitando os campos nome, *e-mail*, matrícula, siape, perfil, administrador e senha. Os campos nome, e-mail e senha são obrigatórios. O campo matrícula é opcional, mas torna-se obrigatório se o perfil for Aluno. Já o campo siape é opcional, mas torna-se obrigatório se o perfil for Professor.

(e) O usuário fornece seus dados e submete o formulário.

(f) O sistema valida os campos informados.

(g) Em caso de sucesso, o registro do novo usuário é incluído no sistema e o usuário é redirecionado para o *login*

(h) Caso contrário, é exibida uma mensagem informando o erro.

3. **UC03 - Cadastrar disciplina**

Ator Coordenador.

Descrição Inclusão de uma disciplina no sistema.

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador acessa o formulário de cadastro de disciplina.

(c) O sistema exibe o formulário de cadastro de disciplina e solicita as informações necessárias para o cadastro: código identificador, nome da disciplina, período letivo, o professor que irá ministrá-la e seu turno.

(d) O coordenador preenche o formulário com os dados da disciplina.

(e) O coordenador submete o formulário.

(f) O sistema valida os dados.

(g) Em caso de sucesso, o sistema inclui o registro da disciplina e redireciona o coordenador para a página de listagem de disciplinas.

(h) Caso contrário, o sistema exibe uma mensagem informando o erro.

4. **UC04 - Editar disciplina**

Ator Coordenador.

Descrição Edição dos dados de uma disciplina cadastrada no sistema

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador seleciona uma disciplina para editar.

(c) O sistema exibe o formulário de edição de disciplina, com seus dados previamente preenchidos.

(d) O coordenador preenche os dados da disciplina que deseja alterar.

(e) O sistema valida os dados editados.

(f) Em caso de sucesso, o sistema registra a edição da disciplina e redireciona o coordenador para a página de listagem de disciplinas.

(g) Caso contrário, o sistema exibe uma mensagem informando o erro.

5. **UC05 - Excluir disciplina**

Ator Coordenador.

Descrição Exclusão do registro de uma disciplina cadastrada no sistema.

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador seleciona uma disciplina para excluir.

(c) O sistema remove o registro da disciplina.

6. **UC06 - Avaliar disciplina**

Ator Aluno.

Descrição Avaliação de uma disciplina cursada pelo aluno durante o período letivo.

Fluxo básico

(a) O aluno se autentica no sistema.

(b) O aluno escolhe uma disciplina para avaliar.

(c) O sistema redireciona para a página de avaliação de disciplina.

(d) O sistema exibe o questionário de avaliação da disciplina.

(e) O aluno responde ao questionário avaliativo.

(f) O aluno submete o questionário.

(g) O sistema valida as respostas.

(h) Em caso de sucesso, o sistema registra a avaliação da disciplina e redireciona o aluno para a página de listagem de disciplinas.

(i) Caso contrário, o sistema exibe uma mensagem informando o erro.

7. **UC07 - Cadastrar período letivo**

Ator Coordenador.

Descrição Inclusão de um período letivo no sistema.

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador navega para a página de listagem de períodos.

(c) O sistema exibe a lista de períodos letivos.

(d) O coordenador acessa a opção de incluir um novo período letivo.

(e) O sistema exibe o formulário de cadastro de período letivo, que solicita o nome e se é o período vigente.

(f) O coordenador informa os dados do período letivo.

(g) O coordenador submete o formulário.

(h) O sistema valida os dados.

(i) Em caso de sucesso, o sistema inclui o registro do período letivo e redireciona o coordenador para a listagem de períodos letivos.

(j) Caso contrário, o sistema exibe uma mensagem informando o erro.

8. **UC08 - Editar período letivo**

Ator Coordenador.

Descrição Edição de um período letivo cadastrado no sistema.

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador navega para a página de listagem de períodos.

(c) O sistema exibe a lista de períodos letivos.

(d) O coordenador escolhe um período letivo para editar.

(e) O sistema exibe o formulário de edição de período letivo.

(f) O coordenador preenche os dados que deseja alterar.

(g) O coordenador submete o formulário.

(h) O sistema valida os dados editados.

(i) Em caso de sucesso, o sistema registra as alterações no período letivo.

9. **UC09 - Excluir período letivo**

Ator Coordenador.

Descrição Exclusão do registro de um período letivo cadastrado no sistema.

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador navega para a página de listagem de períodos.

(c) O sistema exibe a lista de períodos letivos.

(d) O coordenador escolhe um período letivo para excluir.

(e) O sistema remove o registro do período letivo.

10. **UC10 - Visualizar relatórios gerais**

Ator Coordenador.

Descrição Visualização do relatório avaliativo geral.

Fluxo básico

(a) O coordenador se autentica no sistema.

(b) O coordenador navega até a página de relatório.

(c) O sistema exibe os filtros de período letivo e disciplina para os relatórios.

(d) O coordenador seleciona os filtros desejados.

(e) O sistema gera o relatório com base nos filtros preenchidos.

11. **UC11 - Visualizar relatório individual**

Ator Professor.

Descrição Visualização do relatório avaliativo individual.

Fluxo básico

(a) O professor se autentica no sistema.

(b) O professor navega até a página de relatório.

(c) O sistema exibe os filtros de período letivo e disciplinas ministradas para o relatório.

(d) O professor seleciona os filtros desejados.

(e) O sistema gera o relatório com base nos filtros preenchidos.

Os casos de uso listados exemplificam algumas das interações e funcionalidades mais importantes da aplicação.

3.3.6 Diagrama de Entidade-Relacionamento

O Diagrama Entidade-Relacionamento (ER) é usado para representar as entidades, seus atributos e seus relacionamentos, auxiliando na criação do banco de dados e servindo como base para o desenvolvimento do sistema. Assim, é possível analisar a estrutura da aplicação e identificar os relacionamentos entre as entidades.

Na figura 3.4 podemos observar a representação gráfica do Diagrama ER do Sistema de Avaliação de Disciplinas. Nesse diagrama, identificamos que a aplicação possui entidades como Usuário, onde serão armazenados os usuários e seus respectivos perfis (Aluno, Professor ou Coordenador), Disciplina, Avaliação, Período e a tabela intermediária DisciplinaAluno.

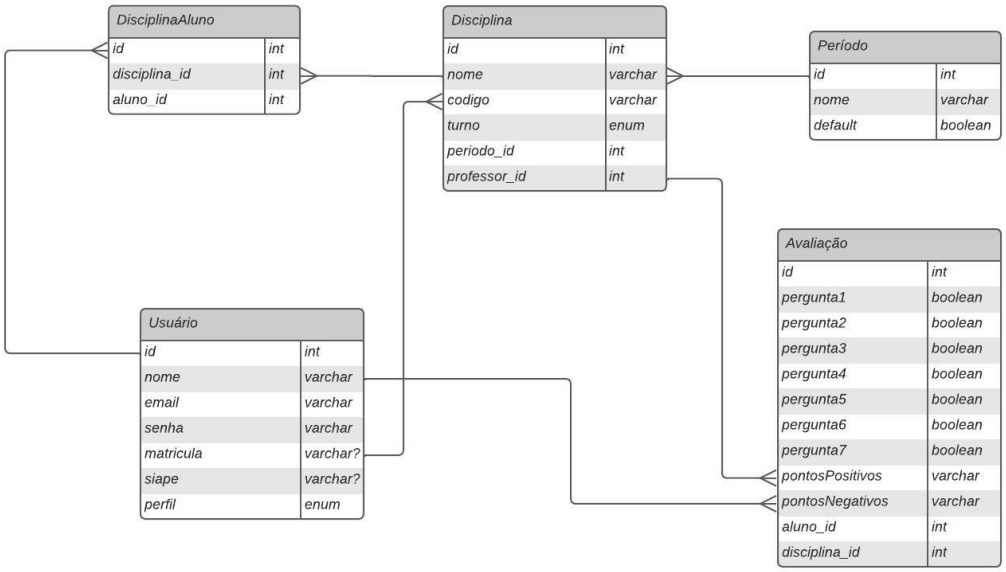

Figura 3.4: Diagrama ER da aplicação proposta

Capítulo 4

Implementação

Neste capítulo será apresentado como foi o desenvolvimento da aplicação proposta. Na seção 4.1 falaremos sobre as tecnologias utilizadas, abordando os *frameworks* escolhidos, o ambiente de execução, o gerenciamento de banco de dados e o ambiente de desenvolvimento. Na seção 4.2 iremos abordar o padrão de arquitetura escolhido e como foi a sua implementação.

4.1 Tecnologias Utilizadas

Para o desenvolvimento da proposta, foi utilizado o *framework* de *frontend* Angular[1] com Bootstrap[2], além do *framework* AdonisJS[3], construído em Node.js[4], no *backend*. Para armazenamento de dados foi escolhido o SQLite[5], um sistema de gerenciamento de banco de dados relacional embutido, gratuito e de código aberto, que oferece integração com o Lucid, o *Object-Relational Mapping* (ORM) utilizado pelo AdonisJS.

[1] <https://angular.io>
[2] <https://getbootstrap.com>
[3] <https://adonisjs.com>
[4] <https://nodejs.org/en>
[5] <https://www.sqlite.org/index.html>

4.1.1 Angular

Angular é um *framework* JavaScript de código aberto escrito em TypeScript e mantido pela Google. Ele é amplamente utilizado para desenvolver aplicações web baseadas em uma única página dinâmica, conhecidas também como SPAs (*Single-Page Applications*). De acordo com Ivanovs (2023), o Angular é o segundo *framework frontend* mais utilizado no mundo desde 2016.

O Angular foi o *framework* escolhido para construir o *frontend* da aplicação por vários motivos. Uma de suas principais vantagens é sua arquitetura modular, que permite que o sistema seja dividido em módulos independentes. Essa abordagem facilita a organização do código e torna-o reutilizável.

Outra vantagem é a sua arquitetura baseada em componentes, que permite criar interfaces interativas e reutilizáveis. Além disso, sua ampla comunidade de desenvolvedores e o suporte contínuo oferecido pela Google são ótimas vantagens de utilizar o *framework*, por facilitar a busca por soluções para desafios encontrados durante o desenvolvimento.

4.1.2 Bootstrap

O Bootstrap é um *framework* gratuito de código aberto utilizado para desenvolver interfaces responsivas. O *framework* oferece uma biblioteca de estilos pré-definidos, com códigos escritos em HTML, CSS e JavaScript. Com as classes disponibilizadas pelo Bootstrap, podemos construir elementos como botões, menus e tabelas. Essas classes são responsivas, possibilitando que a aplicação seja usada em aparelhos com telas de diversos tamanhos.

Algumas vantagens de utilizar o Bootstrap são: responsividade, já que os elementos se adaptam a diferentes tipos de tela, o sistema de *grid*, que divide o conteúdo da tela em linhas e colunas e facilita a criação da responsividade, e a compatibilidade com navegadores, já que o *framework* é compatível com navegadores como Chrome, Edge, Firefox e Safari.

4.1.3 Node.js

O Node.js é um ambiente de execução de JavaScript do lado do servidor, o que significa que é possível executar aplicações Javascript fora do navegador. Ele utiliza o V8, conhecido também como *Chrome's V8 JavaScript engine*, um poderoso interpretador JavaScript desenvolvido pela Google que permite executar o código de forma assíncrona. O Node.js foi desenvolvido em 2009 por Ryan Dahl e, embora seja relativamente novo, é utilizado por grandes empresas como LinkedIn[6], Netflix[7], Uber[8] e Trello[9]. (BREWSTER, 2021)

Dentre as vantagens de utilizar o Node.js estão o modelo I/O não bloqueante e orientado a objetos que permite lidar com uma grande quantidade de chamadas sem gerar bloqueios ou gargalos. Além disso, ele oferece o *Node Package Manager* (NPM), um gerenciador de pacotes que disponibiliza diversos pacotes de código aberto e reutilizável.

4.1.4 AdonisJS

O AdonisJS é um *framework* opinado para Node.js criado em 2015 com o objetivo de fornecer uma estrutura sólida e completa para o desenvolvimento de aplicações web escaláveis.

Dentre as características notáveis do AdonisJS está seu padrão arquitetural *Model-View-Controller* (MVC), que separa o código em modelos, visualizações e controladores, facilitando a manutenção e escalabilidade do código. Além disso, o AdonisJS oferece o Lucid ORM, um mapeador de objetos relacionais que oferece métodos prontos para simplificar a manipulação de banco de dados. O *framework* também oferece recursos para autenticação, o que permite implementar métodos como login e cadastro de forma simples.

[6] <https://www.linkedin.com/>
[7] <https://www.netflix.com/>
[8] <https://www.uber.com/>
[9] <https://trello.com/>

4.1.5 SQLite

O SQLite é um sistema de gerenciamento de banco de dados relacional embutido, gratuito e de código aberto. Diferente de sistemas como PostgreSQL[10] e MySQL[11], o SQLite funciona de forma independente e não necessita de um servidor.

Existem várias vantagens em usar o SQLite. Primeiramente, ele é altamente portável, e está disponível em plataformas como Windows, macOS, Linux, iOS e Android. Além disso, ele é fácil de configurar e não requer instalação, já que o banco de dados é armazenado em um único arquivo. Comparado a outros sistemas de banco de dados, o SQLite possui um tamanho de biblioteca relativamente pequeno, ocupando cerca de 250KB, enquanto servidores como o MySQL podem chegar a 600MB. (ESTRELLA, 2023)

No entando, é importante considerar algumas limitações do uso do SQLite. Dentre elas, a sua escalabilidade limitada, já que a biblioteca não é adequada para altas cargas. Além disso, o bloqueio do banco de dados em operações de gravação simultânea impacta o desempenho em cenários de intensa concorrência.

4.1.6 Ambiente de Desenvolvimento

Para o desenvolvimento da aplicação proposta utilizamos o Visual Studio Code[12], um editor de código-fronte popular e versátil. De acordo com Overflow (2022), o Visual Studio Code foi a ferramenta de edição de texto mais utilizada pelos desenvolvedores em 2022. Para gerenciamento do banco de dados, utilizamos o pacote MySQL[13] do Visual Studio Code, que também oferece suporte para o SQLite. Para versionamento de código, a ferramenta escolhida foi o Git, uma ferramenta de versionamento utilizada por 93% dos desenvolvedores.

[10] <https://www.postgresql.org>
[11] <https://www.mysql.com>
[12] <https://code.visualstudio.com>
[13] <https://dev.mysql.com/doc/visual-studio/en/visual-studio-code-editor.html>

4.2 Arquitetura do Sistema

Para o desenvolvimento do sistema proposto foi utilizado o padrão de arquitetura MVC, uma abordagem que promove a separação de responsabilidades e modularização da aplicação. O padrão MVC divide o sistema em três camadas: *Models* (Modelo), *Views* (Visualização) e *Controller* (Controlador). Nesta seção, iremos detalhar cada uma dessas camadas e descrever como foram implementadas na aplicação proposta.

4.2.1 *Models*

Na arquitetura MVC, a camada *Models* (Modelos) é a responsável por representar os dados e implementar as regras de negócio da aplicação. Ela encapsula os métodos responsáveis por acessar e manipular os dados da aplicação para leitura, criação, exclusão ou atualização. Além disso, a camada de Modelo interage com o banco de dados, realizando consultadas e atualizações, e retorna os dados requisitados pelo Controlador.

No desenvolvimento da aplicação proposta, foram criados os seguintes modelos:

a) **Pessoa** O modelo de Pessoa representa as pessoas cadastradas no sistema. Ele é o responsável por indicar se a pessoa é um Professor, Aluno ou Coordenador. Esse modelo possui uma relação de muitos-para-muitos com o modelo Disciplina, indicando a lista de disciplinas acessíveis pelo Aluno.

b) **Avaliação** O modelo de Avaliação representa as avaliação recebidas pelo sistema. Esse modelo se relaciona com o modelo de Disciplina, para indicar a disciplina avaliada, e com o modelo Pessoa, para indicar o Aluno que realizou a avaliação.

c) **Disciplina** O modelo de Disciplina representa as disciplinas cadastradas no sistema. Esse modelo se relaciona com o modelo de Pessoa para indicar o Professor responsável pela Disciplina, e com o modelo Período, para indicar o período letivo da disciplina.

d) **Período** O modelo de Período representa os períodos letivos cadastrados no sistema. Esse modelo possui uma relação de um-para-muitos com o modelo

de Disciplinas, representando as Disciplinas ofertadas no período letivo.

No código da figura 4.1 é possível visualizar como foi desenvolvido o modelo de Avaliação, e suas relações com o modelo de Pessoa e Disciplina. Os modelos herdam da classe *Model*, desenvolvida pelo Lucid[14], que oferece métodos de manipulação de dados.

```
/** @type {typeof import('@adonisjs/lucid/src/Lucid/Model')} */
const Model = use("Model");

You, seconds ago | 1 author (You)
class Avaliacao extends Model {
  static get table() {
    return "avaliacao";
  }

  pessoa() {
    return this.belongsTo("App/Models/Pessoa", "pessoa_id", "id");
  }

  disciplina() {
    return this.belongsTo("App/Models/Disciplina", "disciplina_id", "id");
  }
}
```

Figura 4.1: Criação do Modelo de Avaliação

4.2.2 *Views*

A camada de *Views*, também conhecida como Visualização, é a responsável por gerar a interface de interação com o usuário. Nessa camada, os dados recebidos do Modelo são processados e apresentados para o usuário da aplicação de maneira interativa. Na arquitetura MVC, a *View* é desacoplada da camada de *Model* e *Controller*, e não possui lógicas de negócio nem interage com a camada de dados.

Para desenvolvimento da camada de Visualização na aplicação proposta utilizamos a biblioteca Bootstrap. A biblioteca disponibiliza classes prontas de CSS, o que facilita a criação de elementos como botões, tabelas e *cards*. Além disso, a biblioteca

[14] <https://docs.adonisjs.com/guides/models/introduction>

oferece alguns recursos interativos como *dropdowns* e modais. Com o Bootstap, podemos gerar interfaces com eficiência para renderizar dados de forma compreensível e intuitiva para os usuários.

Dentre as principais *Views* da aplicação está a tela de login, ilustrada na figura 4.2. Nessa tela, o usuário pode se autenticar no sistema informando o e-mail e a senha. Caso o usuário não esteja cadastro no sistema ele pode clicar no botão **Realizar cadastro**, que irá redirecioná-lo para a tela de cadastro ilustrada na figura 4.3.

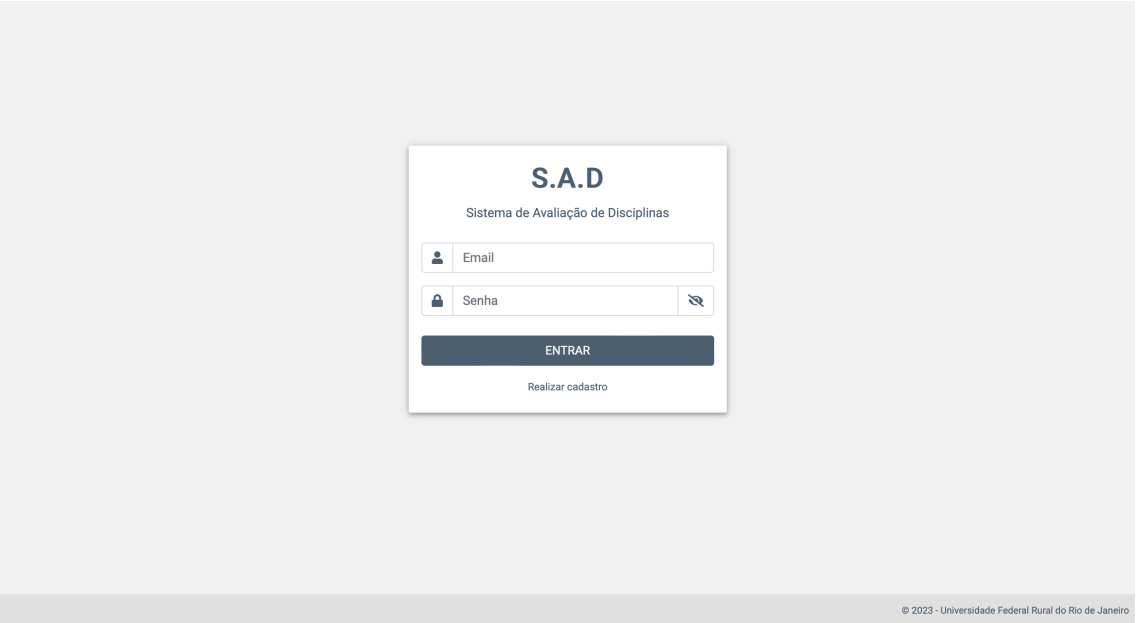

Figura 4.2: Tela de login da aplicação proposta

Se o usuário for um aluno, após se autenticar no sistema ele será redirecionado para a tela de lista de disciplinas como ilustrado na figura 4.4. Nessa tela, o aluno verá quais disciplinas ainda estão pendentes de avaliação. Caso a disciplina já tenha sido avaliada, o botão **Avaliar** estará desabilitado.

Ao clicar no botão **Avaliar** o aluno será redirecionado para a página de avaliar disciplina ilustrada na figura 4.5, onde ele poderá responder o questionário apresentado na subseção 3.3.2 do capítulo anterior.

Caso o usuário seja um professor ele terá acesso à tela de lista de disciplinas, porém verá apenas as disciplinas ministradas por ele. Nessa tela, ilustrada na figura 4.6, o professor poderá visualizar a média das avaliações recebidas para cada um

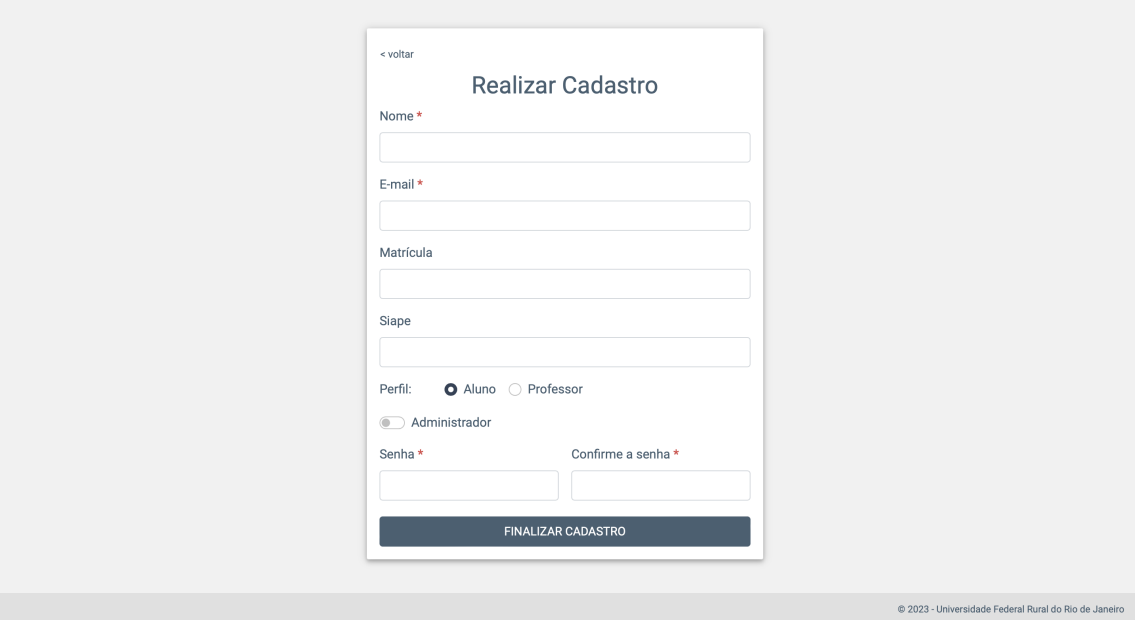

Figura 4.3: Tela de cadastro da aplicação proposta

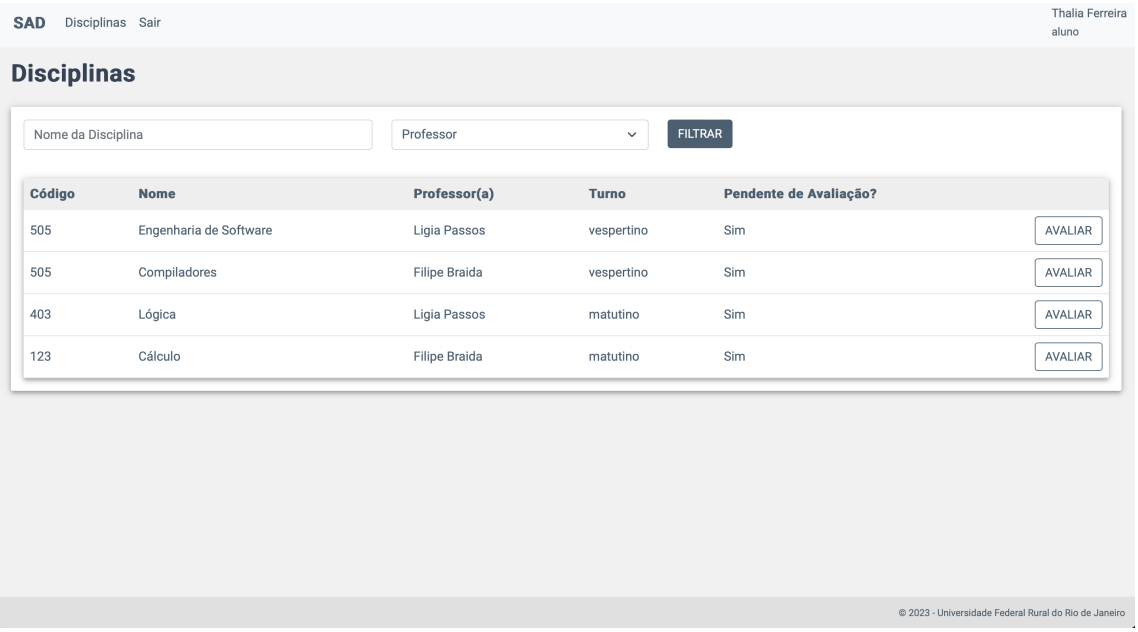

Figura 4.4: Tela de lista de disciplinas do aluno

dos pontos avaliados nas sete questões objetivas do questionário, bem como a média geral de todas as avaliações recebidas para aquela disciplina.

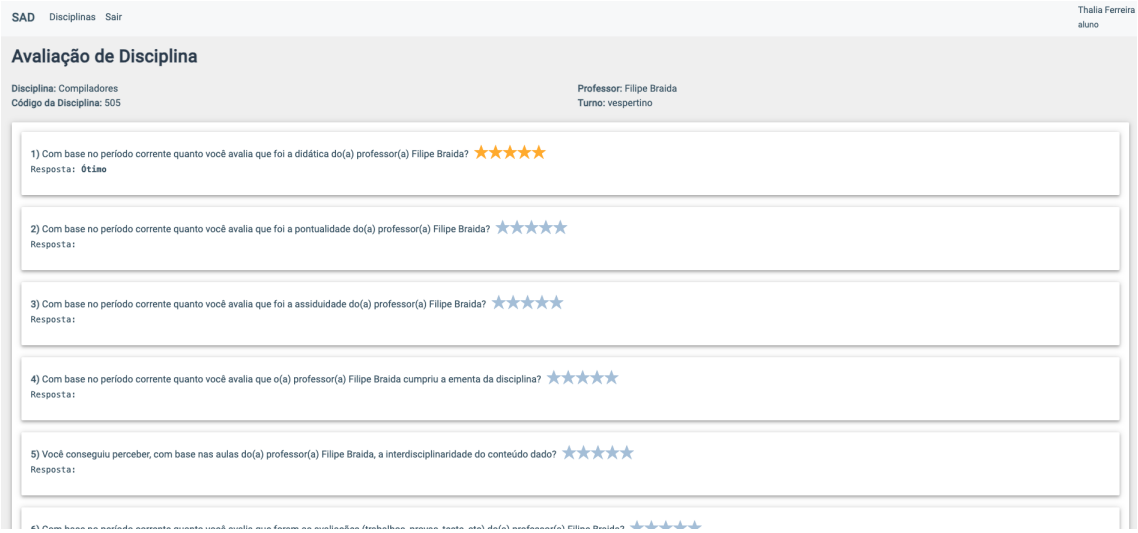

Figura 4.5: Tela de avaliação de disciplina

Figura 4.6: Tela com a média de avaliações das disciplinas do professor

4.2.3 *Controllers*

A camada de *Controllers*, ou Controladores, atua como uma camada intermediária entre as *Views* e os *Models*. Sua principal responsabilidade é interpretar requisições do usuário e coordenar a interação entre as outras camadas. Além disso, as *Controllers* gerenciam todo o fluxo e a lógica da aplicação.

Ao receber uma requisição do usuário, a *Controller* interage com a *Model* para executar as ações solicitadas. Essas ações podem incluir atualizar os dados no *Model*, realizar operações de inserção, ou buscar alguma informação. Após atualizar o *Model*, a *Controller* notifica a *View* para que as telas do sistema sejam atualizadas de acordo

com as alterações realizadas.

Na figura 4.7 é apresentado como foi implementado o método *POST* na *Controller* de Avaliação. O método recebe uma requisição contendo os dados referentes a avaliação realizada. Nele, são feitas as validações para garantir que apenas alunos possam realizar avaliações e que uma disciplina seja avaliada somente uma vez. Após a conclusão das validações, a *Controller* solicita a criação da avaliação ao *Model*.

A presença da camada de *Controllers* é crucial para obter uma arquitetura organizada e modular, facilitando o desenvolvimento e a manutenção da aplicação.

```
async store({ request, response, auth }) {
  try {
    const data = request.only([
      "pergunta1",
      "pergunta2",
      "pergunta3",
      "pergunta4",
      "pergunta5",
      "pergunta6",
      "pergunta7",
      "pontosPositivos",
      "pontosNegativos",
      "disciplina_id",
    ]);

    const pessoa = await Pessoa.findBy("user_id", auth.user.id);

    if (pessoa.perfil != "aluno")
      return response
        .status(500)
        .send("Avaliações devem ser feitas por alunos.");

    const avaliacaoExistente = await Avaliacao.query()
      .where({
        pessoa_id: pessoa.id,
        disciplina_id: data.disciplina_id,
      })
      .first();

    if (avaliacaoExistente)
      return response.status(500).send("Essa disciplina já foi avaliada.");

    const avaliacao = await Avaliacao.create({
      pessoa_id: pessoa.id,
      ...data,
    });
    return avaliacao;
  } catch (error) {
    return response.status(500).send(error.message);
  }
}
```

Figura 4.7: Método de POST na Controller de Avaliação

Capítulo 5

Conclusão

Neste capítulo será apresentada a conclusão do trabalho. Na seção 5.1, apresentaremos um resumo do que foi feito e as considerações finais. Já na seção 5.2, discutiremos sobre as limitações encontradas ao longo do desenvolvimento da proposta, e os possíveis trabalhos futuros.

5.1 Considerações finais

O objetivo deste trabalho foi desenvolver um sistema web de avaliação de disciplinas. Para atingir esse objetivo foram realizadas etapas essenciais, como o levantamento de requisitos do sistema, a definição das regras de negócio e a análise de casos de uso, conforme apresentado na seção 3.3. Durante o desenvolvimento da aplicação, foram utilizadas tecnologias modernas e uma arquitetura bem definida, como descrito no capítulo 4.

Conforme discutido nos capítulos anteriores, permitir que os alunos avaliem as disciplinas cursadas na universidade traz diversos benefícios. Ao implementar uma aplicação web de avaliação, é possível aprimorar o processo de *feedbacks* dos alunos e fornecer informações valiosas ao corpo docente, de maneira prática e eficiente, com o objetivo de promover a qualidade do ensino.

Em conclusão, este trabalho é uma contribuição para aprimorar a transparência e

incentivar a participação ativa dos alunos no ambiente acadêmico, identificar áreas de melhoria no curso e promover discussões e reflexões entre o corpo discente e docente.

5.2 Limitações e trabalhos futuros

A aplicação desenvolvida atinge seu principal objetivo ao permitir que os alunos avaliem as disciplinas cursadas e fornecendo ao corpo docente acesso à média das avaliações recebidas, seguindo todas as principais regras de negócio apresentadas na seção 3.3.3.

Durante o desenvolvimento, no entanto, identificamos uma limitação em relação a como seria realizado o cadastro inicial de alunos, professores e coordenadores, bem como a associação dos alunos e professores às disciplinas. Para resolver esse problema, uma sugestão rápida seria exportar os dados dos usuários do Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA)[1] e importá-los diretamente no banco de dados. Em seguida, seria possível importar as disciplinas e associá-las aos seus respectivos alunos e professores.

Uma das possibilidades de crescimento para a aplicação é a integração da aplicação desenvolvida ao SIGAA, o que traria benefícios significativos como a eliminação da necessidade de um novo cadastro no sistema de avaliação. Os alunos, professores e coordenadores poderiam fazer *login* na aplicação utilizando as mesmas credenciais do SIGAA, garantindo que os usuários estejam associados aos perfis corretos. Com essa integração, não seria necessário realizar importações periódicas diretamente no banco de dados.

Outra possibilidade de melhoria seria o desenvolvimento de um *dashboard* acessível pelos professores e coordenadores do curso. Esse *dashboard* poderá conter gráficos interativos que apresentem dados como: *ranking* das disciplinas avaliadas, evolução das avaliações ao longo do tempo, e uma nuvem de palavras com os comentários recebidos nos campos de pontos positivos e negativos das avaliações.

Também é viável e eficaz aplicar técnicas de *machine learning* para analisar as

[1] <https://sigaa.ufrrj.br/>

opiniões expostas nos campos de texto aberto do questionário. Essa abordagem permitiria diversos tipos de análises, como a extração de termos relevantes, para identificar os principais termos mencionados, e a análise de tendências e padrões, para identificar áreas específicas que requerem melhorias. Também é possível aplicar técnicas de recomendação para recomendar disciplinas optativas de acordo com as tendências expressadas pelos alunos.

Além disso, é possível expandir a aplicação para abranger toda a UFRRJ. Isso permitiria que alunos e professores de todos os cursos tenham acesso ao sistema, possibilitando o cadastro e a avaliação de disciplinas de diferentes áreas além de Ciência da Computação.

Referências

BERNERS-LEE, T. et al. The world-wide web. *Communications of the ACM*, ACM New York, NY, USA, v. 37, n. 8, p. 76–82, 1994.

BERNERS-LEE, T.; FIELDING, R.; FRYSTYK, H. *Hypertext transfer protocol–HTTP/1.0*. [S.l.], 1996.

BERNERS-LEE, T. J. The essence of the web. CERN, 2004.

BRASIL. Lei nº 10.861, de 14 de Abril de 2004. Institui o Sistema Nacional De Avaliação da Educação Superior - SINAES e dá outras providências. *Diário Oficial da União*, 2004. Disponível em: <https://legislacao.presidencia.gov.br/atos/?tipo=LEI&numero=10861&ano=2004&ato=b59Qzaq1UeRpWT347>.

BREWSTER, C. 15 companies that use node.js in 2023 successfully. 2021. Disponível em: <https://www.trio.dev/blog/companies-use-node-js>.

DOMANTAS, G. What is css and how does it work? 2023. Disponível em: <https://www.hostinger.com/tutorials/what-is-css>.

ESTRELLA, C. Sqlite vs mysql – qual a diferença e qual usar. 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/sqlite-vs-mysql>.

FIELDING, R. et al. *Hypertext transfer protocol–HTTP/1.1*. [S.l.], 1999.

HAUBEN, M. History of arpanet. *Site de l'Instituto Superior de Engenharia do Porto*, v. 17, p. 1–20, 2007.

HEMMENDINGER, D. Messaging in the early sdc time-sharing system. *IEEE Annals of the History of Computing*, v. 36, n. 1, p. 52–57, 2014.

IVANOVS, A. The most popular front-end frameworks in 2023. 2023. Disponível em: <https://stackdiary.com/front-end-frameworks/>.

JÚNIOR, N. d. A. Sistema(s) de avaliação da educação superior brasileira. *Cadernos CEDES*, CEDES - Centro de Estudos Educação e Sociedade, v. 29, n. 78, p. 257–269, May 2009. ISSN 0101-3262. Disponível em: <https://doi.org/10.1590/S0101-32622009000200008>.

KILBOURNE, J.; WILLIAMS, T. Unicode, UTF-8, ASCII, and SNOMED CT®. In: AMERICAN MEDICAL INFORMATICS ASSOCIATION. *AMIA Annual Symposium Proceedings*. [S.l.], 2003. v. 2003, p. 892.

KIRAN, H. Javascript usage statistics. 2023. Disponível em: <https://techjury.net/blog/javascript-usage-statistics/#:~:text=As\%20of\%202022\%2C\%2098.7\%25\%20of,language\%2C\%20showing\%20its\%20increasing\%20popularity.>

KLEINROCK, L. An early history of the internet [History of Communications]. *IEEE Communications Magazine*, IEEE, v. 48, n. 8, p. 26–36, 2010.

KROL, E.; HOFFMAN, E. *FYI on"What is the Internet?"*. [S.l.], 1993.

LUKASIK, S. Why the arpanet was built. *IEEE Annals of the History of Computing*, IEEE, v. 33, n. 3, p. 4–21, 2010.

MARILL, T.; ROBERTS, L. G. Toward a cooperative network of time-shared computers. In: *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*. New York, NY, USA: Association for Computing Machinery, 1966. (AFIPS '66 (Fall)), p. 425–431. ISBN 9781450378932. Disponível em: <https://doi.org/10.1145/1464291.1464336>.

MOCKAPETRIS, P. *The domain name system*. [S.l.]: University of Southern California. Information Sciences Institute, 1984.

MOHAMMADI, R.; ESHAGHI, F.; AREFI, M. Internal Evaluation: Appropriate Strategic for Quality Evaluation and Improvement of Management in Departments at Universities(The Case of Iran). *Procedia - Social and Behavioral Sciences*, v. 69, p. 719–728, 2012. ISSN 1877-0428. International Conference on Education Educational Psychology (ICEEPSY 2012). Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877042812054523>.

MOORE, S.; KUOL, N. Students evaluating teachers: exploring the importance of faculty reaction to feedback on teaching. *Teaching in Higher Education*, v. 10:1, p. 57–73, 2005.

OVERFLOW, S. 2022 developer survey. 2022. Disponível em: <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies>.

PHILLIPS, B. Designers: the browser war casualties. *Computer*, IEEE, v. 31, n. 10, p. 14–16, 1998.

POLIDORI, M. M.; MARINHO-ARAUJO, C. M.; BARREYRO, G. B. Sinaes: perspectivas e desafios na avaliação da educação superior brasileira. *Ensaio: Avaliação e Políticas Públicas em Educação*, Fundação CESGRANRIO, v. 14, n. 53, p. 425–436, Oct 2006. ISSN 0104-4036. Disponível em: <https://doi.org/10.1590/S0104-40362006000400002>.

POSTEL, J. *Rfc0821: Simple mail transfer protocol*. [S.l.]: RFC Editor, 1982.

POSTEL, J. *Domain name system structure and delegation*. [S.l.], 1994.

RIABOV, V. V. Smtp (simple mail transfer protocol). *River College*, 2005.

ROBERTS, L. The arpanet and computer networks. In: *A history of personal workstations*. [S.l.: s.n.], 1988. p. 141–172.

SEBENIUS, J. K. Negotiating lessons from the browser wars. *MIT Sloan management review*, 2002.

SETH, S.; VENKATESULU, M. A. *TCP/IP Architecture, Design and Implementation in Linux*. [S.l.]: Wiley New York, NY, USA, 2008.

SOCOLOFSKY, T. J.; KALE, C. J. *TCP/IP tutorial*. [S.l.], 1991.

SUTHERLAND, I. The tx-2 computer and sketchpad. *Lincoln Laboratory Journal*, v. 19, n. 1, p. 82, 2012.