



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**GABRIEL DE SOUSA BARROS**

**EVOLUÇÃO DE SISTEMA DE APOIO PARA DISCIPLINA DE  
LABORATÓRIO DE ESTRUTURA DE DADOS E ALGORITMOS**

**CAMPINA GRANDE - PB**

**2024**

**GABRIEL DE SOUSA BARROS**

**EVOLUÇÃO DE SISTEMA DE APOIO PARA DISCIPLINA DE  
LABORATÓRIO DE ESTRUTURA DE DADOS E ALGORITMOS**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**Orientador : Adalberto Cajueiro de Farias**

**CAMPINA GRANDE - PB**

**2024**

**GABRIEL DE SOUSA BARROS**

**EVOLUÇÃO DE SISTEMA DE APOIO PARA DISCIPLINA DE  
LABORATÓRIO DE ESTRUTURA DE DADOS E ALGORITMOS**

**Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.**

**BANCA EXAMINADORA:**

**Adalberto Cajueiro de Farias**

**Orientador – UASC/CEEI/UFCG**

**Claudio Elízio Calazans Campelo**

**Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro**

**Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 15 de MAIO de 2024**

**CAMPINA GRANDE - PB**

## RESUMO

Na disciplina de Laboratório de Estrutura de Dados e Algoritmos (LEDA) do curso de Ciência da Computação na UFCG, os professores avaliam os alunos através de atividades práticas semanais e provas, também práticas, ao fim de cada unidade do semestre letivo. Dado o grande número de alunos e atividades realizadas, a disciplina utiliza de um sistema para automação de algumas tarefas e facilitação da correção por parte da equipe de avaliação. Esse sistema, apesar de bastante útil, apresenta alguns problemas, estes sendo a falta de privacidade para os dados disponibilizados pelo sistema, uma interface datada e o uso de tecnologias já obsoletas que dificultam a sua manutenção. Este trabalho propõe solucionar estes problemas ao reescrever o sistema utilizando tecnologias mais recentes, atualizando a sua interface e adicionando autenticação e autorização à nova versão. Através dessas adições foi possível verificar de forma qualitativa que o sistema agora oferece maior privacidade aos dados, apresenta uma interface mais agradável e, utilizando de tecnologias mais modernas, facilita a sua manutenção.

# **EVOLUTION OF A SUPPORT SYSTEM FOR DATA STRUCTURE AND ALGORITHM LABORATORY COURSE**

## **ABSTRACT**

In the Data Structures and Algorithms Laboratory discipline of the Computer Science course at UFCG, professors assess students through weekly, practical activities and exams at the end of each semester. Given the large number of students and activities, the course employs a system to automate certain tasks and streamline the grading process for the evaluation team. However, this system, while effective, suffers from several issues: lack of data privacy, an outdated interface, and reliance on obsolete technologies that hinder maintenance.

This project aims to address these problems by rewriting the system using modern technologies, updating its user interface, and adding authentication and authorization to the new version. These enhancements have qualitatively improved the system by ensuring greater data privacy, presenting a more user-friendly interface, and simplifying maintenance through the use of modern technologies.

# Evolução de Sistema de Apoio para Disciplina de Laboratório de Estrutura de Dados e Algoritmos

Gabriel de Sousa Barros  
Universidade Federal de Campina Grande  
Rua Aprígio Veloso, 882  
CEP 58.429-900, Campina Grande, PB, Brasil  
gabriel.barros@ccc.ufcg.edu.br

Adalberto Cajueiro de Farias  
Universidade Federal de Campina Grande  
Rua Aprígio Veloso, 882  
CEP 58.429-900, Campina Grande, PB, Brasil  
adalberto@computacao.ufcg.edu.

## ABSTRACT

In the Data Structures and Algorithms Laboratory discipline of the Computer Science course at UFCG, professors assess students through weekly, practical activities and exams at the end of each semester. Given the large number of students and activities, the course employs a system to automate certain tasks and streamline the grading process for the evaluation team. However, this system, while effective, suffers from several issues: lack of data privacy, an outdated interface, and reliance on obsolete technologies that hinder maintenance.

This project aims to address these problems by rewriting the system using modern technologies, updating its user interface, and adding authentication and authorization to the new version. These enhancements have qualitatively improved the system by ensuring greater data privacy, presenting a more user-friendly interface, and simplifying maintenance through the use of modern technologies.

## RESUMO

Na disciplina de Laboratório de Estrutura de Dados e Algoritmos (LEDA) do curso de Ciência da Computação na UFCG, os professores avaliam os alunos através de atividades práticas semanais e provas, também práticas, ao fim de cada unidade do semestre letivo. Dado o grande número de alunos e atividades realizadas, a disciplina utiliza de um sistema para automação de algumas tarefas e facilitação da correção por parte da equipe de avaliação. Esse sistema, apesar de bastante útil, apresenta alguns problemas, estes sendo a falta de privacidade para os dados disponibilizados pelo sistema, uma interface datada e o uso de tecnologias já obsoletas que dificultam a sua manutenção. Este trabalho propõe solucionar estes problemas ao reescrever o sistema utilizando tecnologias mais recentes, atualizando a sua interface e adicionando autenticação e autorização à nova versão. Através dessas adições foi possível verificar de forma qualitativa que o sistema agora oferece maior privacidade aos dados, apresenta uma interface mais agradável e, utilizando de tecnologias mais modernas, facilita a sua manutenção.

## Palavras-Chave

Aplicação web, atualização, privacidade, manutenibilidade, modernização, interface gráfica

## 1. INTRODUÇÃO

A disciplina “Laboratório de Estrutura de Dados e Algoritmos” (LEDA) do curso de Ciência da Computação da Universidade

Federal de Campina Grande é o complemento prático da disciplina teórica “Estrutura de Dados e Algoritmos” (EDA). A avaliação do desempenho dos alunos é feita a partir de provas práticas realizadas ao fim de cada unidade do semestre letivo e na resolução semanal de roteiros, ambos exigindo a implementação de uma solução para um problema envolvendo o uso de uma estrutura de dados específica ou algoritmo abordados na disciplina teórica. A correção destas atividades é formada por duas partes: a execução de testes automáticos, para validação da funcionalidade esperada da implementação, e a avaliação do *design* da solução desenvolvida pelo aluno, para garantir que a solução esteja dentro dos padrões estabelecidos pela disciplina.

Para que essa metodologia funcione, é necessário um ambiente que permita a submissão das soluções desenvolvidas, a execução automatizada dos testes e o acesso ao código dos roteiros submetidos para a avaliação do *design*. Pela falta de um sistema que abrangesse todas as demandas da disciplina, um sistema próprio foi desenvolvido e é utilizado desde 2013.

No entanto, o sistema apresenta algumas limitações. Primeiramente, qualquer usuário pode realizar o *download* das submissões das atividades, enquanto que o ideal seria apenas os professores, monitores e o aluno que submeteu o código tivessem acesso. Além disso, a interface de usuário do sistema precisa de melhorias que a tornem mais intuitiva para uma melhor experiência de usuário. E, por fim, o sistema foi desenvolvido em uma tecnologia já obsoleta, dificultando sua manutenção.

Em função disso o objetivo deste trabalho é o desenvolvimento de um novo sistema utilizando a arquitetura Cliente-Servidor. Assim, atualizando as tecnologias utilizadas, oferecendo uma interface aprimorada e garantindo uma melhor privacidade aos alunos que submeterem seus trabalhos realizados durante a disciplina.

## 2. PROBLEMAS

A equipe da disciplina relatou que o sistema em uso apresenta problemas nos seguintes tópicos:

1. Tecnologia utilizada: o sistema foi construído utilizando-se de uma tecnologia já obsoleta, dificultando a manutenção do mesmo;
2. Privacidade: o sistema só restringe o acesso à funcionalidade de correção das atividades realizadas, deixando todas as outras informações disponíveis para qualquer usuário que o acesse;

3. Interface de usuário: o sistema apresenta uma aparência antiga, necessitando de modernização para que se alcance uma melhor experiência para o usuário.

Além desses, também foi observado um problema de autenticidade dos laboratórios submetidos. Devido a falta de um mecanismo para identificar quem está fazendo a submissão no envio das atividades, qualquer usuário pode realizar um envio em nome de um aluno. Porém, após discussão com a equipe da disciplina, foi decidido que esse problema não se encontra no escopo deste trabalho por limitação de tempo.

### 3. ARQUITETURA DA SOLUÇÃO

O sistema consiste em uma aplicação *web* com o objetivo de auxiliar na execução da disciplina de Laboratório de Estrutura de Dados e Algoritmos, tanto em relação ao trabalho realizado pelos professores como no fornecimento de informações de interesse aos alunos matriculados. Permitindo assim o cadastro, acesso, submissão e correção das atividades avaliativas necessárias para a realização da disciplina. Cada ação tendo seu acesso restrito a depender do tipo de usuário acessando o sistema.

#### 3.1 Visão Geral

Junto à decisão de atualizar as tecnologias utilizadas no sistema também veio a necessidade de mudar a arquitetura para que seguisse o modelo Cliente-Servidor[1]. Neste modelo há uma divisão de responsabilidades em que o servidor assegura a lógica de negócio, armazena e fornece os dados necessários para o funcionamento da aplicação, e o cliente requisita esses dados do servidor e os apresenta de forma legível ao usuário. A comunicação entre essas partes do sistema foi feita utilizando o padrão REST nas requisições entre um e outro. Antes do início do trabalho algumas partes do sistema já haviam sido implementadas, o servidor estava num estado muito próximo ao atual e o cliente já tinha uma base desenvolvida. O trabalho atual teve um foco principalmente na resolução dos problemas de privacidade.

#### 3.2 Servidor

O servidor atualmente utilizado foi desenvolvido na linguagem Java, utilizando do *framework* Spring e da ferramenta Spring Boot[2]. O *framework* é conhecido por oferecer ao desenvolvedor ferramentas e recursos que facilitam a criação de aplicações com baixo acoplamento e capazes de serem executadas em ambientes variados. Já a ferramenta Spring Boot é responsável por facilitar o uso do Spring por oferecer os seguintes benefícios: automação no processo de configuração do aplicativo, sugestões de dependências com base nas necessidades do projeto e a possibilidade de executar a aplicação de forma independente sem a necessidade de um servidor web externo a ela.

A persistência dos dados é feita a partir de planilhas Excel visando uma maior simplicidade no *deployment* da aplicação e uma integração mais fluida com os dados externos necessários para seu funcionamento. Os dados dos alunos matriculados, por exemplo, são fornecidos pelo Controle Acadêmico nesse mesmo formato. Ademais, como as submissões são todas baseadas em arquivos, o *back-end* armazena seus dados no próprio sistema de arquivos, pois isso permite uma manipulação direta na recuperação e na correção automática dos roteiros.

### 3.3 Cliente

A parte do cliente foi desenvolvida utilizando a *framework* Angular[3] na versão 14 e a biblioteca de componentes de interface de usuário Angular Material.

Angular é uma *framework* de front-end baseada em Typescript mantida por uma equipe, de mesmo nome, da empresa Google. A ferramenta possibilita a criação de componentes reutilizáveis, facilitando e acelerando o desenvolvimento de interfaces gráficas ao reduzir a repetição de trabalho.

Já o Angular Material é uma biblioteca que fornece vários componentes Angular estilizados e com suas respectivas funcionalidades já implementadas. São componentes como botões, tabelas, inputs desenvolvidos para que sua aparência siga um mesmo tema e sejam flexíveis para que o desenvolvedor possa personalizá-los, adequando-os ao contexto em que forem utilizados. Dessa forma facilitando o desenvolvimento de uma interface coesa e diminuindo o tempo necessário para o desenvolvimento do projeto.

### 3.4 Funcionalidades do sistema

Nesta seção será apresentado o funcionamento do sistema, detalhando as funcionalidades implementadas. O sistema apresenta uma barra lateral utilizada para a navegação entre as páginas e um header onde é possível selecionar o semestre letivo que deseja-se visualizar e um botão para que o usuário possa se autenticar no sistema.

#### 3.4.1 Submissão das atividades

As soluções para as atividades realizadas na disciplina são implementadas utilizando a linguagem de programação Java. Com isso em mente, um *plugin* Maven previamente desenvolvido é utilizado para o envio do código desenvolvido pelo aluno cursando a disciplina. O *plugin*, após ser configurado com a identificação do aluno pela sua matrícula, realiza o trabalho de compactar os arquivos referentes à solução implementada e os envia ao servidor. O servidor então realiza testes automatizados na solução enviada e armazena os resultados que é utilizado posteriormente para formação da nota do aluno.



FIGURA 1 - Tela de listagem de alunos

#### 3.4.2 Listagem dos alunos matriculados

Na FIGURA 1 é possível observar a exibição de uma lista dos alunos matriculados na disciplina separados por turma





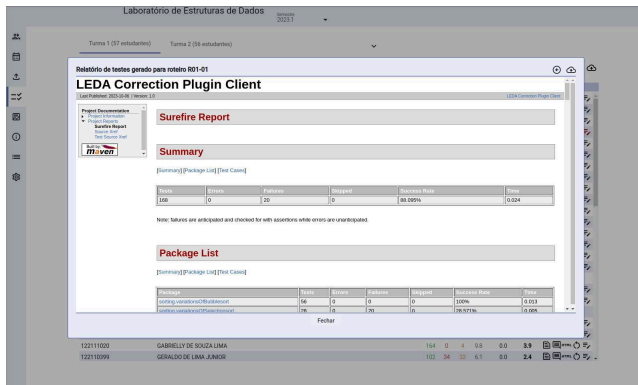


FIGURA 6 - Tela de relatório de testes

### 3.4.6 Correção de Atividades

Na figura 7 é apresentada a interface de correção de roteiros. Nela é fornecido ao monitor corretor o código fonte da solução submetida pelo aluno e o resultado dos testes automatizados. Para a correção é necessário informar o quanto a solução se adequa ao que foi requisitado pela atividade (variando entre baixa e alta) e a classificação qualidade do *design* da solução (variando entre ruim e excelente) e oferecendo as opções de falta e presença respectivamente para os casos de entrega vazia e completa falta da entrega). Além disso, o monitor tem a opção de cadastrar um comentário, que pode conter justificativas para a nota atribuída à atividade ou indicações de possíveis melhorias para a solução apresentada.

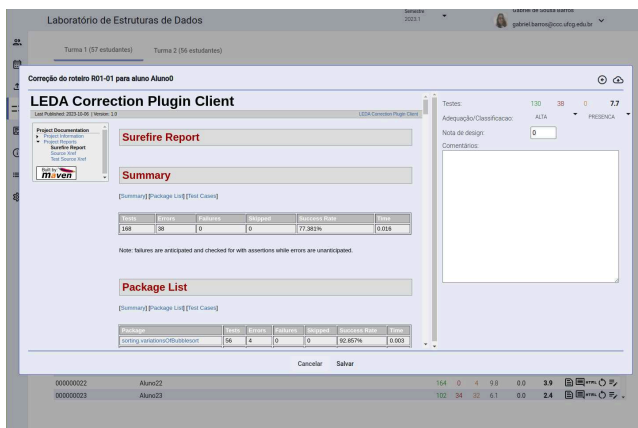


FIGURA 7 - Tela de Correção de atividades

### 3.4.7 Relatório de notas

O sistema oferece uma visualização geral das notas de cada turma, como apresentado na Figura 8, informando os resultados de cada atividade e prova realizada e a média final delas. Cores são utilizadas para classificar a média como “aprovação”, com a cor verde, e “reprovação”, com a cor vermelha.

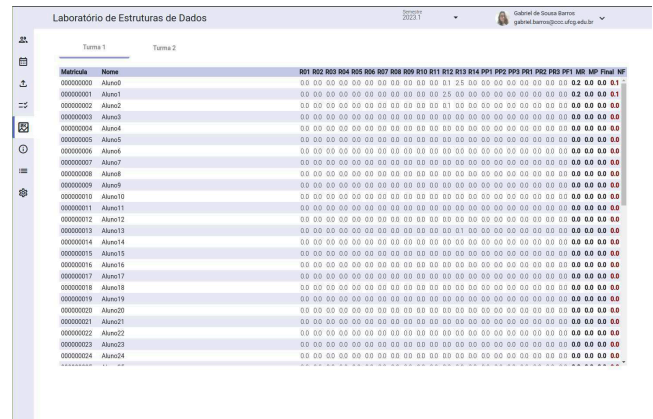


FIGURA 8 - Tela de relatório de notas

### 3.4.8 Prazos de envio de atividades

A Figura 9 representa a visualização dos períodos em que as atividades são disponibilizadas e o prazo final para o envio das mesmas. Apresentando, também, o identificador da atividade, seu nome e a coluna “Ambiente” informando se o *download* dela está disponível.

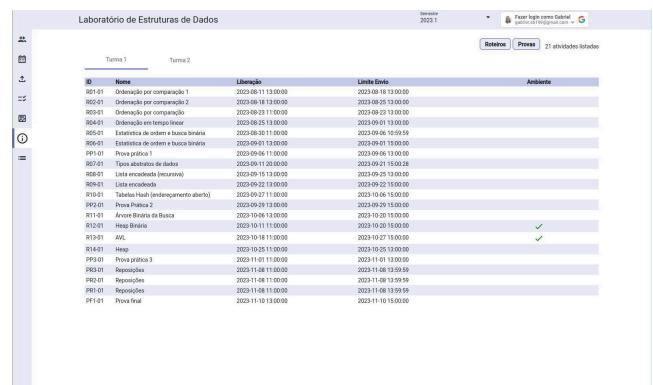


FIGURA 9 - Tela de prazos de envio de atividades

### 3.4.9 Configurações do sistema

Uma página para a configuração do sistema é disponibilizada para os professores ajustarem algumas características do sistema. A aba “Usuários” visualizada na Figura 10 dá a capacidade de desativar alunos que, por algum motivo, deixem de fazer parte da turma naquele semestre, definir na coluna “role” o papel de um usuário ou editar as suas informações. Já a aba “Cronograma” permite a alteração do cronograma exibido na aplicação.

Matricula	Nome	Email	Fone	Role	Turno	Estado	Ações
1000000001	GABRIEL DE SOUSA BARROS	gabriel.barros@ufcg.edu.br	00000-0000	PROFESSOR	1	<input type="checkbox"/>	<input type="checkbox"/>
000000002	Aluno2			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000001	Aluno1			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000002	Aluno2			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000003	Aluno3			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000004	Aluno4			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000005	Aluno5			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000006	Aluno6			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000007	Aluno7			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000008	Aluno8			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000009	Aluno9			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000010	Aluno10			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000011	Aluno11			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000012	Aluno12			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000013	Aluno13			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000014	Aluno14			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000015	Aluno15			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000016	Aluno16			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000017	Aluno17			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000018	Aluno18			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000019	Aluno19			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>
000000020	Aluno20			ALUNO	1	<input type="checkbox"/>	<input type="checkbox"/>

FIGURA 10 - Tela de configuração do sistema

### 3.5 Privacidade

Para que o sistema funcione bem e atinja seu objetivo de facilitar a execução da disciplina a qual ele se atrela é necessário que exista um controle de acesso das funcionalidades e recursos disponibilizados por ele. Para isso, primeiro é necessária uma forma de identificar os usuários utilizando o sistema

#### 3.5.1 Autenticação

Foi implementado um sistema de autenticação utilizando o método de Single Sign-On (SSO)[4] do Google, associando os usuários do sistemas aos seus emails institucionais, criados pela Unidade Acadêmica de Sistemas e Computação. Nesse método, delegamos ao Google a lógica de autenticação, e é possível usufruir de um *token* final que representa a identidade do usuário autenticado. Essa identidade carrega a informação do e-mail e nome do usuário que efetuou o *login*, o que permite identificar se o usuário é um aluno da universidade e se está matriculado na disciplina. Essa verificação cruzada é feita nas planilhas de alunos mencionadas anteriormente e mantidas pelo servidor. Após a verificação o usuário Google é associado à matrícula do aluno.

Essa abordagem traz diversas vantagens para a aplicação como:

- Diminuição do trabalho necessário para a implementação da autenticação do sistema para apenas a checagem da validade do *token* enviado em cada requisição.
- Melhor experiência para o usuário por não ser necessária a criação de uma conta específica para o sistema para acessá-lo.
- Fornece uma maneira fácil de identificar quem está realizando o login através do nome e email fornecidos pelo sistema de SSO.

Os demais usuários que acessam o sistema sem fazer *login* (ou os que fazem *login* mas não estão matriculados) possuem visões restritas do sistema.

#### 3.5.2 Autorização

Além de dar o acesso a algumas funcionalidades apenas após a autenticação do usuário, também foi necessário restringi-las para o tipo de usuário utilizando o sistema. Para isso foi implementado o *Role Based Access Control* (RBAC) [5], também conhecido como sistema de controle de acesso baseado em papéis. Esse

sistema consiste no agrupamento dos usuários a partir dos papéis desempenhados por eles, definidos durante o desenvolvimento, e na associação das permissões de acesso a esses papéis.

No sistema os papéis definidos e suas respectivas permissões foram:

- **Visitante:** um usuário que não efetuou o login. Pode apenas acessar informações como o cronograma e as notas das atividades.
- **Aluno:** aluno matriculado na disciplina no semestre vigente. Pode realizar o download e a submissão das atividades avaliativas realizadas na disciplina.
- **Monitor Corretor:** Monitores da disciplina. Quando atribuídos à correção de um roteiro podem acessar as submissões dos alunos e efetuar a correção, deixando também comentários.
- **Professor:** Possui acesso total ao sistema. Incluindo a configuração das informações exibidas como o cronograma, notas das atividades, e a atribuição dos monitores às correções dos roteiros

## 4. IMPLEMENTAÇÃO

### 4.1 Metodologia de Desenvolvimento

O sistema foi desenvolvido em um trabalho conjunto entre o professor e os monitores atuando na disciplina. Devido a pouca disponibilidade de tempo para a contribuição ao projeto por parte dos monitores e ao contato constante durante as aulas, foi decidido que não seria necessário o uso de uma metodologia de desenvolvimento bem definida. Sendo assim, o processo decorreu de maneira mais informal, tendo as funcionalidades necessárias e a priorização de atividades discutidas durante a atuação dos monitores nas aulas da disciplina.

### 4.2 Principais Desafios

Os desafios enfrentados durante a realização do trabalho foram relacionados ao uso de uma framework front-end não utilizada anteriormente. Para que fosse possível contribuir para o projeto foi necessário um tempo de estudo para compreender as características e o funcionamento do Angular, sendo esses: a construção de seus componentes e a comunicação entre eles, a atualização de seus estados e seus gatilhos para renderizar novamente os componentes exibidos em tela.

## 5. AVALIAÇÃO DA SOLUÇÃO

Para avaliar que a solução resolve de maneira satisfatória os problemas apresentados, ela será, a seguir, discutida de forma qualitativa, destacando as alterações realizadas e seu papel em solucionar os problemas.

O problema do uso de tecnologias obsoletas foi resolvido pelo desenvolvimento do novo sistema utilizando Angular 14 e Java com Spring, tecnologias atuais e com ampla utilização no contexto acadêmico da UFCG.

As falhas em relação à privacidade foram solucionadas pela implementação dos sistemas de autenticação e autorização. Juntos

sendo capazes de restringir o acesso aos dados oferecidos pelo sistema para que sejam acessíveis apenas aos usuários adequados.

A interface de usuário foi modernizada através do uso da biblioteca de componentes Angular Material, que permitiu com maior facilidade o desenvolvimento de uma interface coesa. Porém, ainda são necessários alguns aprimoramentos para que o sistema se adapte a telas menores.

## 5.1 Trabalhos Futuros

Com base na avaliação e nos aspectos do sistema observados durante o desenvolvimento propõe-se que seja possível submeter as atividades através da aplicação web, assim seria possível associar o envio ao usuário logado no sistema, diminuindo bastante a possibilidade que um usuário submeta uma solução de atividade em nome de outra pessoa.

Outra melhoria seria a melhor responsividade do layout do sistema em telas menores. Isso tornará muito mais agradável a experiência de uso da aplicação em celulares e outros dispositivos.

## 6. REFERÊNCIAS

- [1] Wikipédia, a enciclopédia livre, Modelo Cliente-Servidor.  
Disponível em:  
[https://pt.wikipedia.org/wiki/Modelo\\_cliente-servidor](https://pt.wikipedia.org/wiki/Modelo_cliente-servidor)
- [2] O que é Java Spring Boot? Disponível em:  
<https://www.ibm.com/br-pt/topics/java-spring-boot>
- [3] Wikipédia, a enciclopédia livre, Angular (framework).  
Disponível em:  
[https://pt.wikipedia.org/wiki/Angular\\_\(framework\)](https://pt.wikipedia.org/wiki/Angular_(framework))

[4] O que é SSO (Single sign-on)? Disponível em:  
<https://www.ibm.com/br-pt/topics/single-sign-on>

[5] Role-Based Access Control | Ultimate Guide. Disponível em:  
<https://budibase.com/blog/app-building/role-based-access-control/>

## 7. AGRADECIMENTOS

Gostaria de dedicar esse espaço para expressar a minha profunda gratidão a todas as pessoas que contribuíram de alguma forma para que a realização deste trabalho se tornasse possível

Agradeço imensamente à minha família, especialmente aos meus pais, por desde a minha infância apoiarem e incentivarem a minha formação acadêmica.

Ao meu orientador pelo suporte e direcionamento essenciais para o desenvolvimento do projeto.

E, por fim, aos meus amigos pela ajuda e companhia que tornaram essa jornada muito mais agradável e proveitosa.