



UNIVERSIDADE ESTADUAL PAULISTA
“JULIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS - CÂMPUS DE
BAURU



Trabalho de Conclusão de Curso
Bacharelado em Sistemas de Informação

Musicalis - Sistema ERP Educacional

João Vitor Garcia da Silva

José Remo Ferreira Brega

Bauru-SP
2024

S586m

Silva, João Vitor

Musicalis : Sistema ERP Educacional / João Vitor Silva. -- Bauru,
2024

29 p. : il.

Trabalho de conclusão de curso (Bacharelado - Sistemas de
Informação) - Universidade Estadual Paulista (UNESP), Faculdade de
Ciências, Bauru

Orientador: José Remo Ferreira Brega

1. ERP. 2. Desenvolvimento de Software. 3. Otimização via
Software. I. Título.

Agradecimentos

Gostaria de agradecer a todos os meus familiares, à minha namorada e a todos os meus amigos que me proporcionaram as experiências que me trouxeram até aqui.

"A prática é o critério da verdade."

Autor desconhecido

Resumo

Documentação do processo de desenvolvimento de um Sistema ERP Web utilizando Java, Angular, MySQL e outras tecnologias, com foco educacional e que será doado para o projeto da Banda e Orquestra Municipais de Bauru.

Possui o objetivo de automatizar e otimizar processos administrativos atualmente manuais, além de centralizar e padronizar informações.

Palavras-chave: ERP, Desenvolvimento de Software, Otimização

Abstract

Documentation of the development process for a Web ERP System using Java, Angular, MySQL, and other technologies, with an educational focus, to be donated to the Municipal Band and Orchestra project in Bauru. The goal is to automate and optimize currently manual administrative processes, as well as centralize and standardize information.

Keywords: ERP, Software Development, Otimization

Sumário

1	Introdução	8
1.1	Motivação	8
1.2	Soluções Existentes	8
1.3	Comparativo	11
2	Desenvolvimento	13
2.1	Engenharia de Requisitos e Funcionalidades	13
2.2	Tecnologias escolhidas	14
3	Arquitetura	15
3.1	Visão de alto nível	15
3.2	Expandindo a visão da API Java	16
3.3	Servidor	18
3.4	Segurança	18
3.5	Melhorias futuras	19
4	Interface	20
5	Conclusão	28
	Referências	29

Lista de Figuras

1	Página inicial no site do Emusys[2]	9
2	Rolando um pouco mais abaixo, podemos ver algumas funcionalidades[2]	9
3	Página inicial no site do WebMusica[6]	10
4	Página inicial no site do SisMusica[5]	10
5	Um pouco mais abaixo, podemos ver os planos de pagamento[5]	11
6	Diagrama de Contêineres	15
7	Diagrama de Componentes - Expandindo API	16
8	Representação do fluxo entre Aplicação Web - Controlador - Serviço - Repositório - Banco de dados	17
9	Representação dos Contêineres Docker dentro do Servidor	18
10	Página inicial da versão de demonstração do template Sakai, onde pode ser visualizado um exemplo de <i>Dashboard</i>	20
11	Página inicial do PrimeNG, ao lado direito podem ser visualizados alguns dos componentes disponíveis	20
12	Página inicial, menu lateral e Dashboard	21
13	Demonstração do cadastro de alunos	21

14	Demonstração da listagem de alunos	21
15	Acompanhamento de Notas de um bimestre	22
16	Visualização de uma turma, onde alunos podem ser gerenciados através de botões ou funcionalidade <i>drag and drop</i>	22
17	Agendamento de Avaliações	23
18	Listagem de Avaliações	23
19	Ao colocar sua matrícula, um aluno pode visualizar as datas disponíveis que seu professor agendou	24
20	Mensagem de sucesso após se vincular a uma avaliação	25
21	Avaliação já não aparece mais como livre, e consta o aluno vinculado	25
23	Visualização de uma avaliação finalizada	26
22	Registro da avaliação com a opção de finalização selecionada, onde o professor poderá adicionar a nota da banca e visualizar a performance do aluno nesse bimestre, que contará para sua média final	26
24	Avaliação agora consta como finalizada na listagem	27

1 Introdução

1.1 Motivação

A Banda Sinfônica e Orquestra Sinfônica Municipal de Bauru foram fundadas pelas Leis 4.861/2002 e Lei 5140/2004 respectivamente. São gerenciadas e coordenadas pela Secretaria de Cultura e mantida pela Prefeitura de Bauru. Formada por jovens estudantes entre 11 e 25 anos, a Banda e Orquestra têm os objetivos de promover o estudo técnico musical e agregar valores fundamentais na formação dos integrantes, estimulando o trabalho em grupo, cooperação, integração, autoafirmação, disciplina e respeito entre os seus componentes.

Atualmente, a Banda e a Orquestra atendem mais de 160 crianças e adolescentes promovendo estudo musical com aulas práticas e teóricas gratuitas, sem necessidade de que os alunos adquiram instrumentos[1].

Como membro da Banda Sinfônica Municipal de Bauru por mais de 7 anos, pude acompanhar de perto, através da ótica de aluno, diversos processos como: agendamento e realização de provas práticas, acompanhamento de aulas e controle de alunos. A implementação de um sistema de gerenciamento ajuda na concentração e padronização da informação, na facilidade de acesso e na sua segurança. Além do aumento do conforto e velocidade de processos através de sua automatização.

O Musicalis é Sistema ERP Web de gerenciamento acadêmico que poderá ser acessado por qualquer computador ou smartphone, inicialmente através do navegador, com acesso a internet. Esse sistema está em constante evolução, e é capaz de realizar funcionalidades *Create, Read, Update, Delete (CRUD)* de Alunos, Professores, Turmas, Avaliações e Patrimônios, assim padronizando tanto o formato das informações como o seu acesso, além de outras funcionalidades.

Ele será doado para o projeto da Banda e Orquestra Municipal de Bauru sem nenhum custo de desenvolvimento.

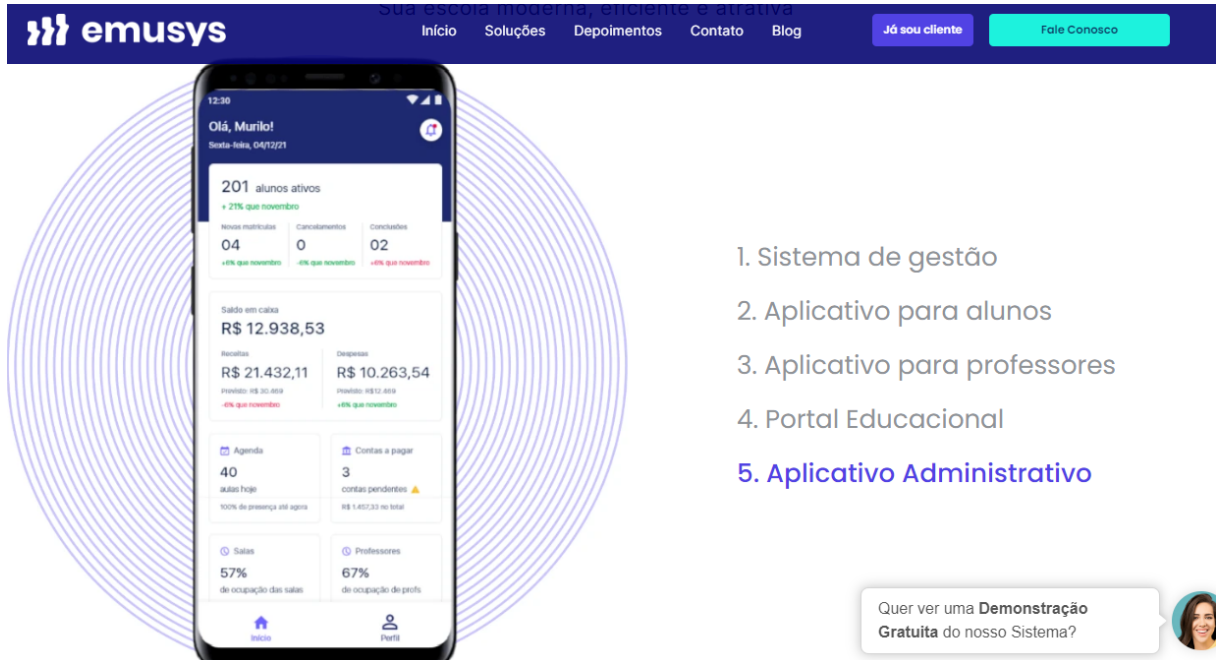
1.2 Soluções Existentes

Nesta seção são apresentadas algumas das soluções encontradas atualmente que têm como objetivo a gestão educacional com foco no nicho musical.

Emusys



Figura 1: Página inicial no site do Emusys[2]



1. Sistema de gestão
2. Aplicativo para alunos
3. Aplicativo para professores
4. Portal Educacional
5. Aplicativo Administrativo

Figura 2: Rolando um pouco mais abaixo, podemos ver algumas funcionalidades[2]

WebMusica

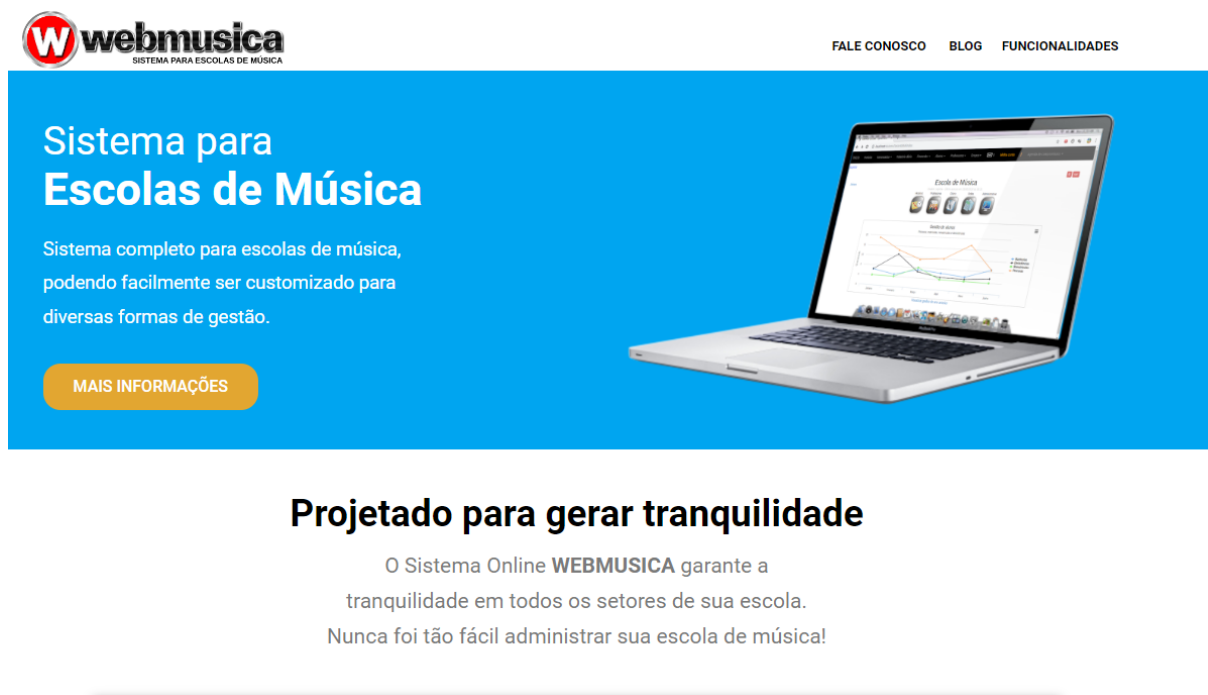


Figura 3: Página inicial no site do WebMusica[6]

SisMusica

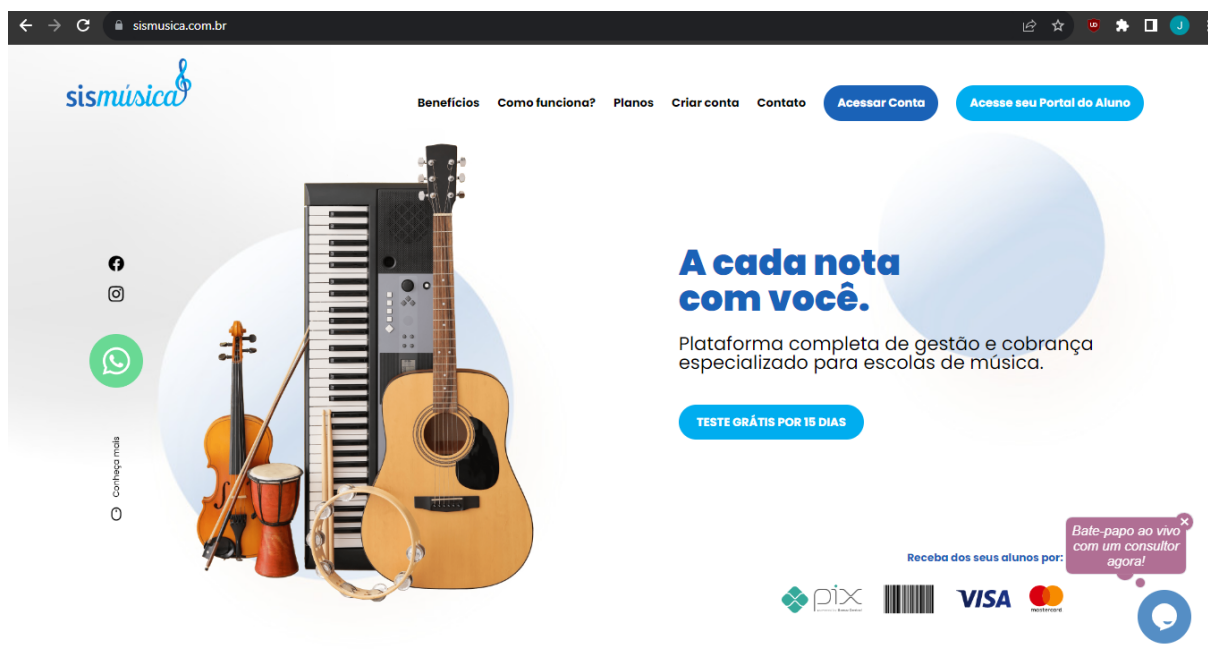


Figura 4: Página inicial no site do SisMusica[5]



Figura 5: Um pouco mais abaixo, podemos ver os planos de pagamento[5]

O objetivo do Musicalis é servir como gerenciamento educacional musical assim como estas soluções, porém como um software personalizado, ao invés de uma solução padrão.

Uma das desvantagens dos softwares padrão em comparação com o software personalizado, é que existe a chance de existirem muitas funcionalidades além das necessárias, o que pode resultar em uma pior experiência para o usuário final. O excesso de funcionalidades pode causar falta de espaço e tempo excessivo de treinamento, além de sensações de desorganização visual e fragilidade do sistema[8].

1.3 Comparativo

Soluções padrão:

Prós:

- Grande quantidade de funcionalidades, capazes de atender muitos clientes diferentes.
- Podem ser acessados em diversos dispositivos.
- Serviço de suporte.

Contras:

- Podem ocasionar em falta de clareza visual e confiabilidade caso o cliente necessite somente de poucas funcionalidades.
- Custo de assinatura alto.
- Possível alta de funcionalidades específicas.

Musicalis

Prós:

- Produzido sob demanda de acordo com as necessidades.
- Pode ser acessado em diversos dispositivos.
- Custo apenas para manutenção, sem perspectiva de lucro.

Contras:

- Extremamente nichado.
- Sem serviço de suporte dedicado.
- Difícil de ser aplicado em outros clientes sem uma alta manutenção/alteração.

Grande parte do apelo do Musicalis é, além de ser personalizado para o projeto, ter um preço de custo que visa somente a manutenção, sendo possível contratar um serviço de servidor extremamente barato que sirva somente as necessidades de processamento do sistema, podendo ser diversas vezes mais barato do que as soluções analisadas.

Um exemplo de funcionalidade que não foi encontrada nos exemplos analisados é o controle de patrimônio. Cada instrumento do projeto possui um número de patrimônio na prefeitura, e quando um instrumento é emprestado a um aluno para que ele possa executar as aulas, é interessante manter um controle sobre estes empréstimos. Como no contexto das soluções analisadas o aluno teria seu próprio instrumento, ou não ficaria responsável por ele por tempo indeterminado, essa funcionalidade não é necessária.

Porém, como está sendo desenvolvido por somente uma pessoa, Musicalis não contará com um serviço de suporte profissionalizado e amplamente disponível, contando apenas com contato direto ao desenvolvedor. O que será mitigado pela sua simplicidade e facilidade de treinamento do usuário.

Junto com essa personalização e nicho, está incluída a dificuldade em vender esse software como um produto para outra instituição ou cliente, sem que haja as devidas modificações para torná-lo atrativo caso as funcionalidades já existentes não sejam necessárias ou suficientes, porém esse não é o objetivo do software inicialmente.

2 Desenvolvimento

2.1 Engenharia de Requisitos e Funcionalidades

Em conjunto ao conhecimento adquirido durante os anos de participação do projeto, foi realizada uma reunião presencial com grande parte do corpo docente do projeto, contando com Professores e Maestro. Nesta reunião foram levantados vários requisitos, dentre eles:

- Histórico de alunos e professores;
- Gerenciamento do Patrimônio;
- Capacidade de filtrar os alunos por idade;
- Acompanhamento do estado de conservação do patrimônio;
- Visualização de um Manual do aluno;
- Integração com uma página pública institucional;
- Gerador de declarações de comparecimento ou de viagem;
- Link que aponta para o diário oficial de quando o aluno começou a fazer parte do projeto;

Dado o levantamento de todos estes requisitos, foi tomada a liberdade de escolher quais são elegíveis para formar o MVP (Mínimo Produto Viável), estes foram:

- Cadastro de Alunos
- Cadastro de Professores
- Cadastro de Turmas
- Cadastro de Diários de aula
- Gerenciamento de Patrimônios
- Agendamento de Avaliações práticas
- Acompanhamento de Notas
- Sistema de Login
- Acesso por múltiplos dispositivos

2.2 Tecnologias escolhidas

Angular

Angular é um *framework* criado e mantido pela Google, que auxilia no desenvolvimento de aplicações baseadas em web. Foi escolhido para a aplicação por ser uma tecnologia robusta e prática, possibilitando escalabilidade e confiança.

Java

Java é uma das linguagens de programação mais utilizadas do mundo até hoje, criada em 1995 pela Sun Microsystems [7].

Spring

Spring é um *framework* que auxilia no desenvolvimento de aplicações Java utilizando conceitos como Inversão de Controle e Injeção de Dependências além de módulos que facilitam o desenvolvimento web tornando a aplicação mais legível e aumentando sua manutenibilidade.

MySql

Simple e completo sistema de gerenciamento de banco de dados relacional gratuito e de código livre, amplamente utilizado.

Docker

Plataforma que oferece virtualização a nível de sistema operacional gratuitamente, permitindo que programas sejam executados em ambientes virtuais.

Hostinger

Provedor de servidor escolhido para a versão inicial do projeto, com opção de hospedagem no Brasil para menor latência.

NGINX

Servidor de aplicação que é utilizado para distribuir e permitir o acesso à aplicação Angular, além de servir como proxy reverso.

Let's Encrypt

Autoridade em certificados que não visa lucro, no qual foi possível adquirir gratuitamente a certificação HTTPS

3 Arquitetura

3.1 Visão de alto nível

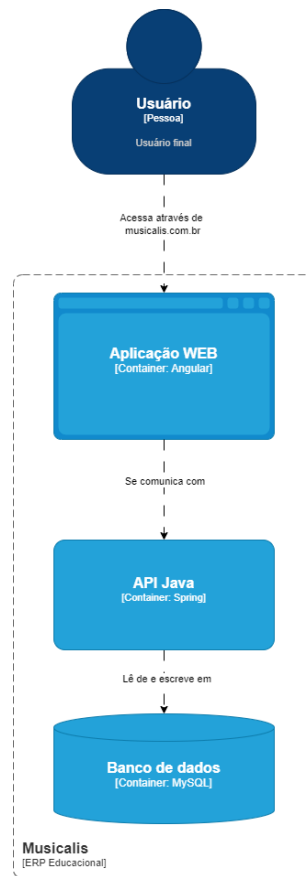


Figura 6: Diagrama de Contêineres

De um ponto de vista mais abstrato e simplificado, a aplicação é composta por uma Aplicação WEB em Angular, que consome uma API Java através de requisições HTTP, que por sua vez escreve e lê informações de um banco de dados MySQL.

3.2 Expandindo a visão da API Java

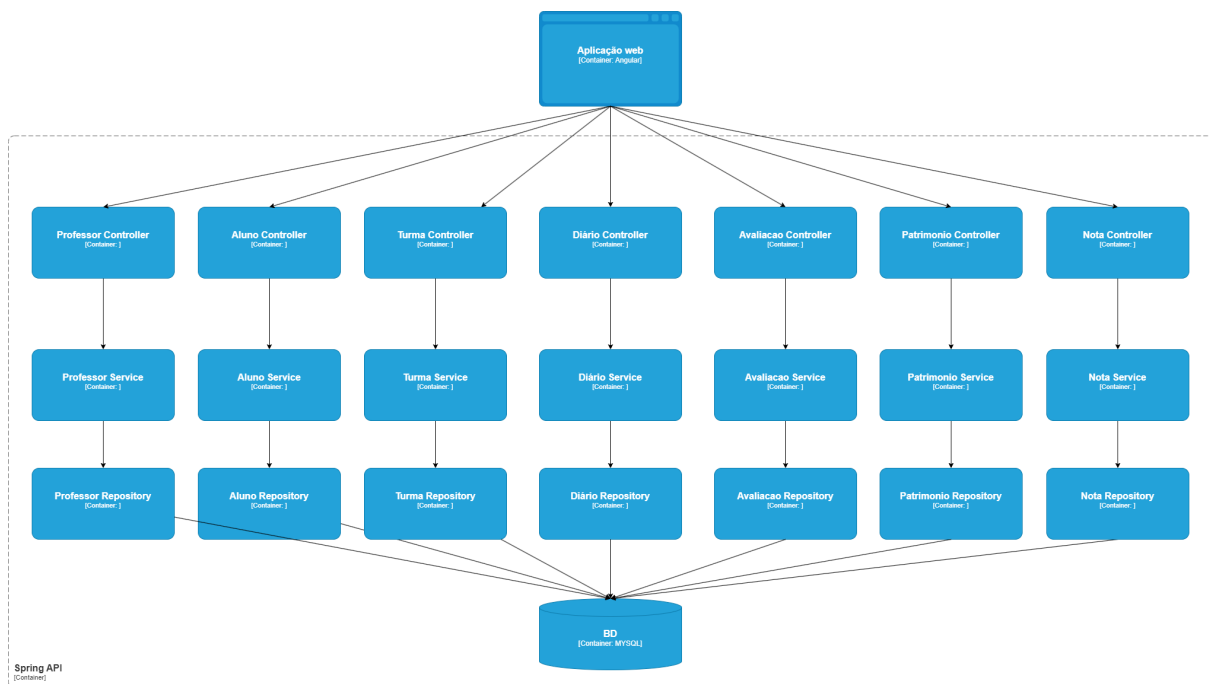


Figura 7: Diagrama de Componentes - Expandindo API

Visão da Arquitetura da API Java mostrando a utilização dos Padrões de Design Controlador, Serviço e Repositório, que tem como objetivo aumentar a manutenibilidade e a clareza do código .

"Problemas tendem a se repetir várias vezes no desenvolvimento de software. Então, são abstraídas soluções para estes problemas, que consistem em um conjunto de técnicas informais que podem ser reutilizadas, estas técnicas são padrões de design"[9] (tradução nossa¹)

O Controlador vem do padrão *MVC* (*Model View Controller*), que resumidamente consiste na separação das responsabilidades entre Modelo (entidade utilizada que é manipulada), Controlador (que o usuário utiliza para manipular o modelo) e Visão (que é atualizada pelo Modelo com as informações). Nesse caso, a Visão pode ser abstraída como a aplicação web, o Modelo como as entidades do sistema que são manipuladas e enviadas através de requisições HTTP como um arquivo JSON, e o Controlador é somente a interface que recebe os pedidos de manipulação do usuário através destas requisições, como por exemplo operações de busca, alteração ou exclusão de registros.

No Controlador estão concentrados todos os métodos que recebem requisições HTTP. Após isso, o Controlador apenas repassa essas ordens para o Serviço.

O Serviço tem como objetivo concentrar a lógica de negócio, ou seja, todas as manipulações e operações necessárias nos dados durante a execução das funcionalidades do

¹Problems tend to recur, and as web programmers we must solve them time and time again. [...] informal set of techniques that we use and reuse in our projects. These techniques are patterns of design.

sistema. Para a execução destas, e a manipulação das informações, é necessário lê-las ou escrevê-las no banco de dados, e para isto o serviço utiliza o Repositório.

O Repositório concentra somente os métodos que interagem com o banco de dados, seja lendo, escrevendo ou apagando registros.



Figura 8: Representação do fluxo entre Aplicação Web - Controlador - Serviço - Repositório - Banco de dados

Dessa forma, existe uma clara separação de responsabilidades entre Controlador, Serviço e Repositório, assim facilitando a manutenção posterior do código, além de sua reutilização (um serviço pode consumir um método de outro, por exemplo).

3.3 Servidor

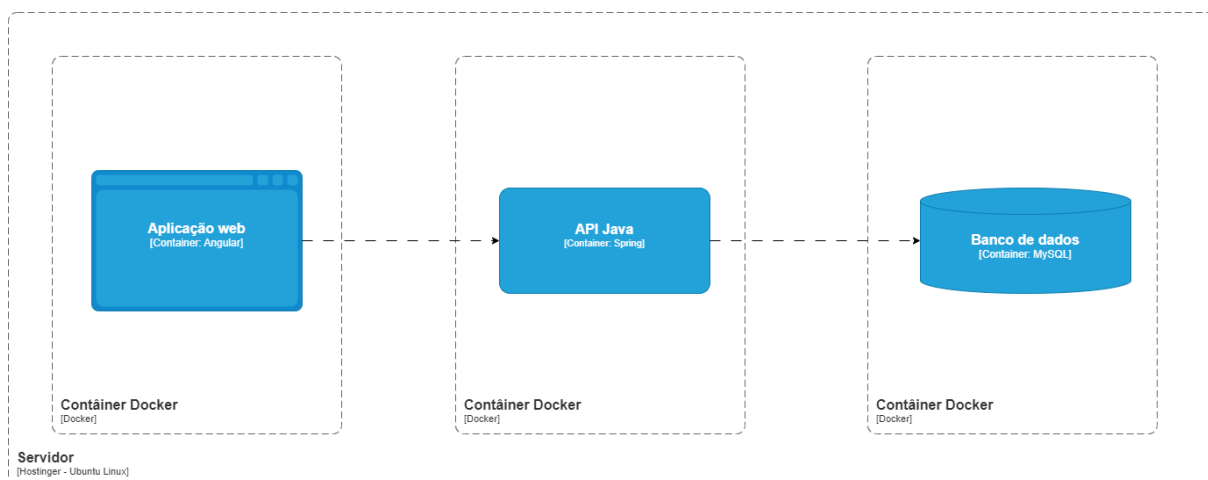


Figura 9: Representação dos Contêineres Docker dentro do Servidor

Cada um destes componentes apresentados anteriormente estão rodando em um servidor adquirido através da Hostinger, com o sistema operacional Ubuntu Linux. Dentro deste servidor existem Contêineres Docker separados para cada uma das partes da aplicação, que se comunicam entre si.

Um contêiner é um pacote leve e reutilizável de software que contém tudo necessário para rodar uma aplicação, desde seu código até suas dependências. Como cada um desses softwares está contido em seu contêiner, possibilita que eles sejam executados em qualquer dispositivo capaz de executar o Docker, e apenas iniciando-o, a aplicação estará funcionando. Do contrário, seria necessário instalar todas as dependências necessárias de cada um destes softwares no dispositivo, ocasionando em maior custo para configuração, além do risco de incompatibilidades e erros.

3.4 Segurança

O domínio 'musicalis.com.br' também adquirido pela Hostinger, redireciona para a aplicação através de um Proxy Reverso utilizando Nginx, de maneira a manter oculto o endereço IP e as portas da aplicação para dificultar ataques, além de permitir uma experiência mais clara ao usuário, que deverá acessar o domínio ao invés do número do IP em seu navegador.

Além deste, existe um tem como objetivo também mascarar o IP e a porta do servidor, porém em outro contexto. Quando um usuário acessa o sistema, ele está rodando em seu navegador a aplicação web, e esta por sua vez faz pedidos para a API através de requisições HTTP que devem ser enviadas para o endereço IP do servidor, na porta correspondente.

Para mascarar esta porta e o IP, o proxy reverso está configurado para encaminhar requisições recebidas no domínio "musicalis.com.br/api/" para o endereço interno correspondente.

3.5 Melhorias futuras

Além de todas as funcionalidades citadas, o processo do desenvolvimento de software é sempre contínuo e iterativo. Logo, existem funcionalidades planejadas para o futuro mesmo antes da apresentação do software para o projeto para validação.

- O acesso ao sistema deverá ser vetado a autenticação via *login* de usuários previamente cadastrados.
- Token de autenticação do sistema web gerado e verificado pelo módulo Spring Security do Spring.
- Todas as senhas deverão ser criptografadas.
- Um usuário não-administrador tem acesso somente a registros que ele mesmo criou.
- Um usuário não autenticado deverá ser redirecionado para a página de login automaticamente.
- Expansão das funcionalidades para os alunos e não só o corpo docente.
- Desenvolvimento de uma aplicação Android, para que o sistema não necessite ser acessado através do navegador em dispositivos móveis.

4 Interface

Foi utilizado um template gratuito e de uso livre chamado Sakai[4], além da biblioteca PrimeNG[3] que disponibiliza componentes gráficos gratuitamente, ambos desenvolvidos pela PrimeTek.

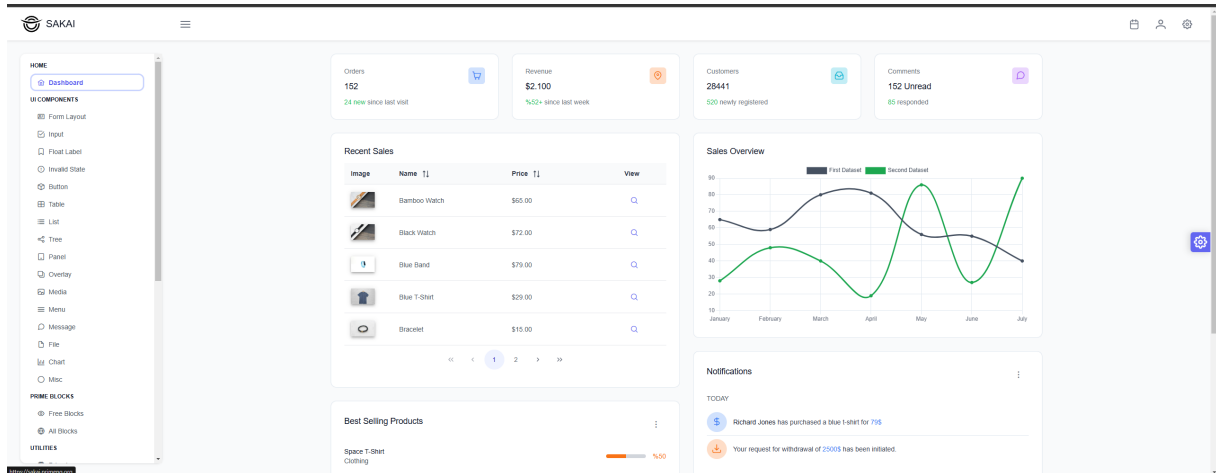


Figura 10: Página inicial da versão de demonstração do template Sakai, onde pode ser visualizado um exemplo de *Dashboard*

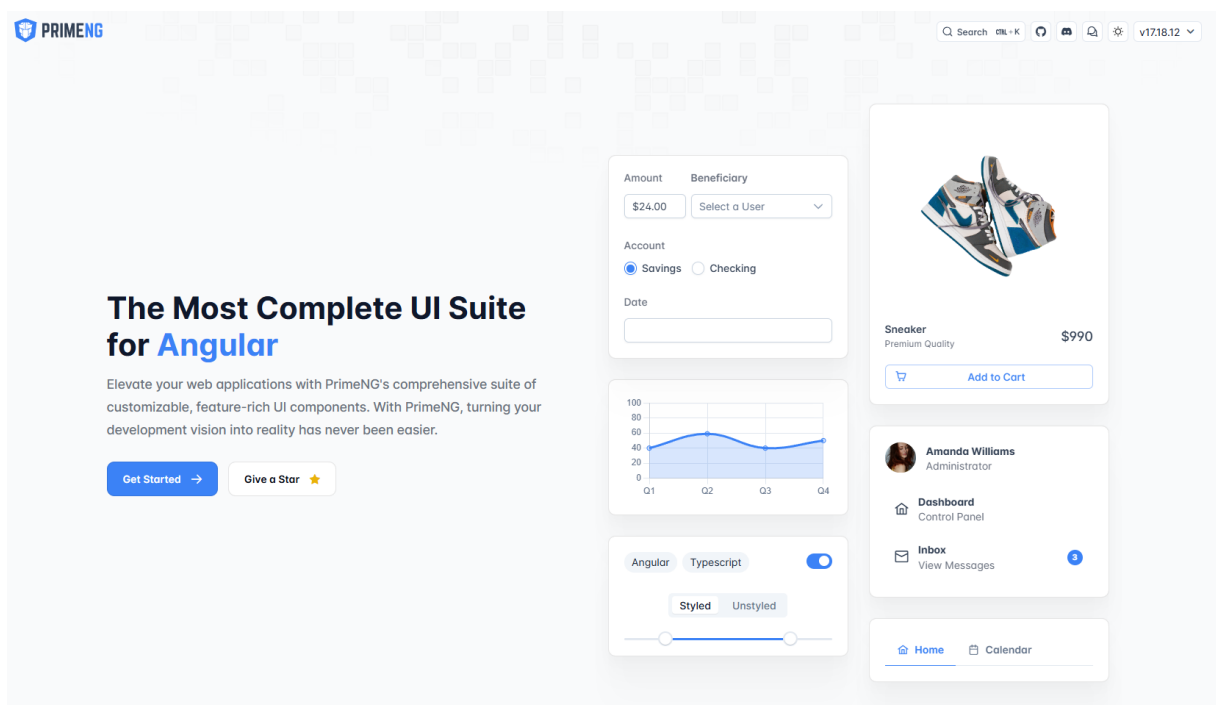


Figura 11: Página inicial do PrimeNG, ao lado direito podem ser visualizados alguns dos componentes disponíveis

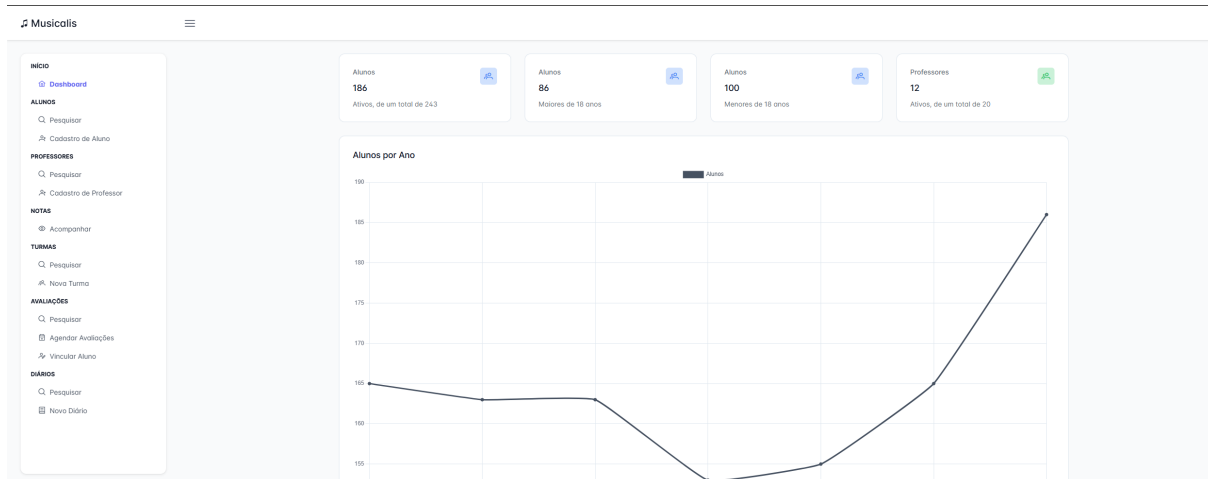


Figura 12: Página inicial, menu lateral e Dashboard

The 'Cadastrar Aluno' form includes the following fields: 'Nome' (text input), 'RG' (text input), 'CPF' (text input), 'Matricula' (text input), 'Data de Nascimento' (text input), 'Instrumento' (dropdown menu), and 'Status' (toggle switch). Below the form is a blue 'Cadastrar' button.

Figura 13: Demonstração do cadastro de alunos

Matricula ↑↓ ▾	Nome ↑↓ ▾	Instrumento ↑↓ ▾	Idade ↑↓ ▾	Status ↑↓ ▾	Ações
12342	joao da silva	Trombone	23	✓ ATIVO	✎ 🗑️
123422	joao	Eufonio	23	✗ INATIVO	✎ 🗑️
11342	joao da silva	trombone	23	✗ INATIVO	✎ 🗑️
12312	João Vitor Garcia	Trompete	21	✗ INATIVO	✎ 🗑️
322222	teste	Oboe	29	✓ ATIVO	✎ 🗑️
2222	Teste da Silva	Trompete	19	✓ ATIVO	✎ 🗑️

Mostrando 1 - 6 de 6 registros << < 1 > >> 10 ▾

Figura 14: Demonstração da listagem de alunos

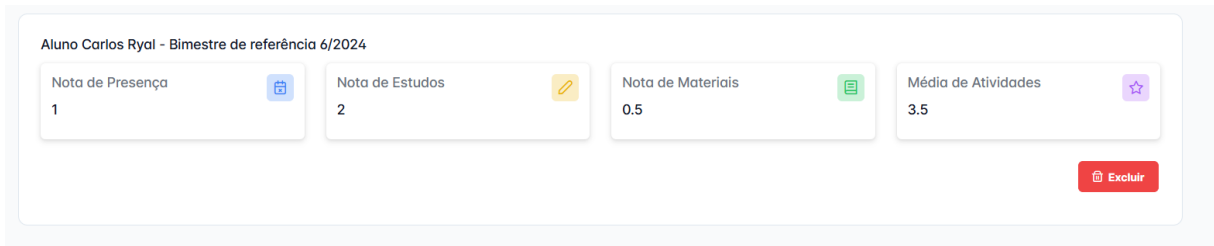


Figura 15: Acompanhamento de Notas de um bimestre

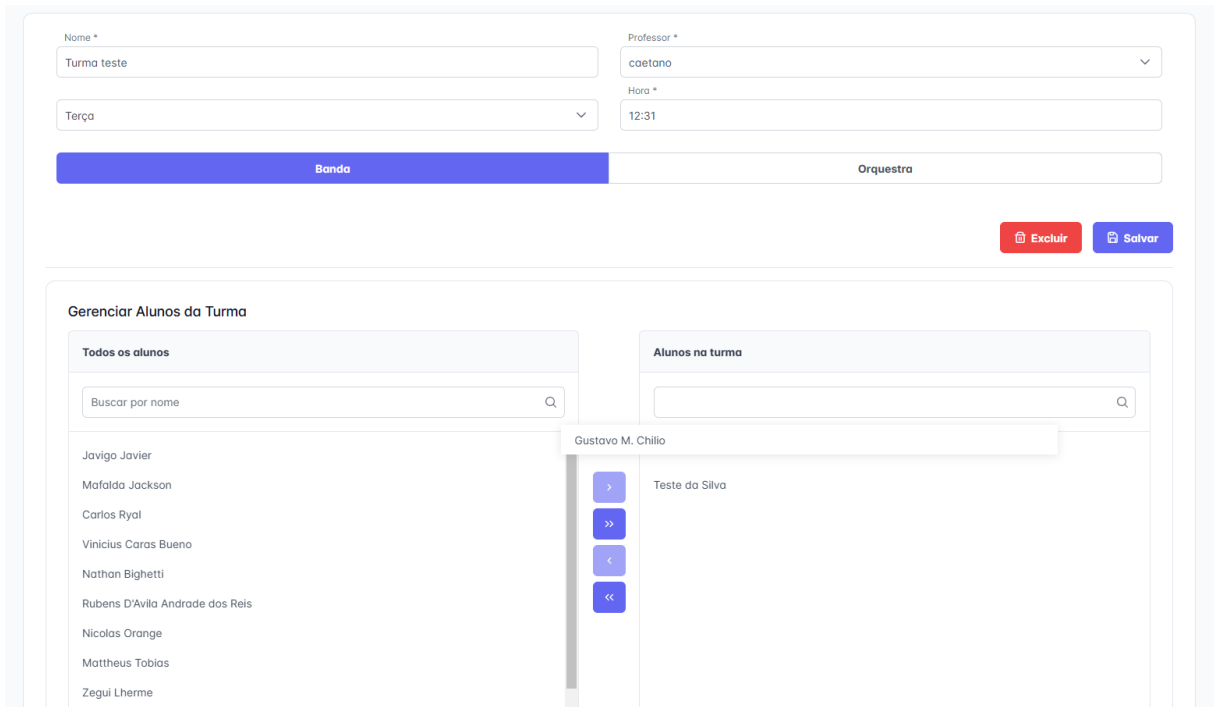


Figura 16: Visualização de uma turma, onde alunos podem ser gerenciados através de botões ou funcionalidade *drag and drop*

Uma das funcionalidades é o fluxo completo de agendamento e realização de avaliações práticas. Ao invés de preencher uma lista com as datas e horários e replicá-la no grupo de Whatsapp do projeto, os professores agora podem executar um agendamento de diversas avaliações de uma só vez, escolhendo a banca presente, a data e o horário.

Figura 17: Agendamento de Avaliações

Após isso, essas avaliações podem ser visualizadas na listagem, e caso não tenham um aluno vinculado, aparecem como "Prova Livre"

Aluno	Professor	Data	Finalizada	Grupo
Prova Livre	Teste	25-11-2024 10:43	NÃO	Banda
Prova Livre	Teste	26-11-2024 10:43	NÃO	Banda
Teste da Silva	caetano	27-11-2024 10:44	NÃO	Banda
Teste da Silva	Tiago Oliveira Tigres	26-11-2024 11:23	SIM	Banda
Apresentação Da Silva	Tiago Oliveira Tigres	27-11-2024 11:23	NÃO	Banda
Prova Livre	Tiago Oliveira Tigres	28-11-2024 11:23	NÃO	Banda
Prova Livre	Felipe Lobo	06-12-2024 11:00	NÃO	Banda
Prova Livre	Felipe Lobo	06-12-2024 12:00	NÃO	Banda
Prova Livre	Felipe Lobo	06-12-2024 12:45	NÃO	Banda

Mostrando 1 - 9 de 9 registros

Figura 18: Listagem de Avaliações

Um aluno pode ser manualmente vinculado a uma avaliação por um usuário, mas também pode ser vinculado pelo próprio através de um acesso externo.

Marcar Avaliação

333222

Aluno Teste da Silva - Clarinete

Professor(a) Felipe Lobo

Data ↑↓
06-12-2024 11:00
06-12-2024 12:00
06-12-2024 12:45

<< < 1 > >>

Figura 19: Ao colocar sua matrícula, um aluno pode visualizar as datas disponíveis que seu professor agendou

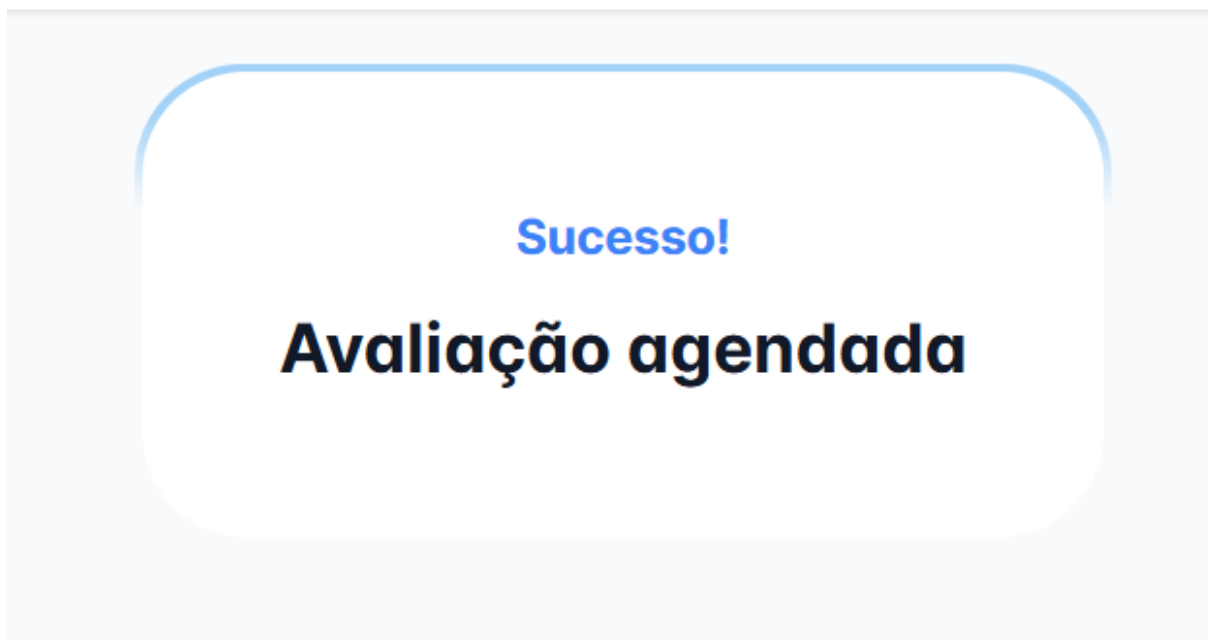


Figura 20: Mensagem de sucesso após se vincular a uma avaliação

A screenshot of a table showing evaluation records. The table has columns for Aluno, Professor, Data, Finalizada, and Grupo. A "Limpar" button is at the top left. The "Finalizada" column contains status indicators: "X NÃO" (red) and "SIM" (green). The table shows 9 records, with the first one being "Prova Livre" by "Teste" on 25-11-2024.

Aluno ↑↓ ▾	Professor ↑↓ ▾	Data ↑↓ ▾	Finalizada ↑↓ ▾	Grupo ↑↓ ▾
Prova Livre	Teste	25-11-2024 10:43	X NÃO	Banda
Prova Livre	Teste	26-11-2024 10:43	X NÃO	Banda
Teste da Silva	coetano	27-11-2024 10:44	X NÃO	Banda
Teste da Silva	Tiago Oliveira Tigres	26-11-2024 11:23	✓ SIM	Banda
Apresentação Da Silva	Tiago Oliveira Tigres	27-11-2024 11:23	X NÃO	Banda
Prova Livre	Tiago Oliveira Tigres	28-11-2024 11:23	X NÃO	Banda
Prova Livre	Felipe Lobo	06-12-2024 11:00	X NÃO	Banda
Teste da Silva	Felipe Lobo	06-12-2024 12:00	X NÃO	Banda
Prova Livre	Felipe Lobo	06-12-2024 12:45	X NÃO	Banda

Mostrando 1 - 9 de 9 registros << < 1 > >> 10 ▾

Figura 21: Avaliação já não aparece mais como livre, e consta o aluno vinculado

No momento da realização da avaliação, o professor pode acessar o registro onde terá a opção de finalizá-la.



Figura 23: Visualização de uma avaliação finalizada

Professor *
Felipe Lobo

Aluno *
Teste da Silva

Avaliador 1 *
Tiago Oliveira Tigres

Avaliador 2 *
Marcelo F. Souza

06/12/2024 12:00

Banda | Orquestra

Finalizar avaliação?

Nota Professor *

Nota Avaliador 1 *

Nota Avaliador 2 *

Nota de Presença: 0.6

Nota de Estudos: 1

Nota de Materiais: 0.28

Média de Atividades: 1.88

Figura 22: Registro da avaliação com a opção de finalização selecionada, onde o professor poderá adicionar a nota da banca e visualizar a performance do aluno nesse bimestre, que contará para sua média final

Após a finalização da avaliação, o histórico permanece, mostrando a nota do aluno no bimestre referente, as notas da banca e sua média final.

[🗑️ Limpar](#)

Aluno ↑↓ ▾	Professor ↑↓ ▾	Data ↑↓ ▾	Finalizada ↑↓ ▾	Grupo ↑↓ ▾
Prova Livre	Teste	25-11-2024 10:43	✗ NÃO	Banda
Prova Livre	Teste	26-11-2024 10:43	✗ NÃO	Banda
Teste da Silva	caetano	27-11-2024 10:44	✗ NÃO	Banda
Teste da Silva	Tiago Oliveira Tigres	26-11-2024 11:23	✓ SIM	Banda
Apresentação Da Silva	Tiago Oliveira Tigres	27-11-2024 11:23	✗ NÃO	Banda
Prova Livre	Tiago Oliveira Tigres	28-11-2024 11:23	✗ NÃO	Banda
Prova Livre	Felipe Lobo	06-12-2024 11:00	✗ NÃO	Banda
Teste da Silva	Felipe Lobo	06-12-2024 12:00	✓ SIM	Banda
Prova Livre	Felipe Lobo	06-12-2024 12:45	✗ NÃO	Banda

Mostrando 1 - 9 de 9 registros << < 1 > >> 10 ▾

Figura 24: Avaliação agora consta como finalizada na listagem

5 Conclusão

Com foco na criação de um software que atenda as necessidades específicas da Banda e Orquestra Sinfônica Municipal de Bauru, foi desenvolvido um sistema ERP Web utilizando Java, Angular, MySQL e Docker. Com canal aberto com os coordenadores e professores dos projetos, foram identificados os principais requisitos a serem atendidos. Assim, durante o processo de engenharia de requisitos foram traçadas como podem ser automatizadas as etapas da rotina de aulas, e também, como poderiam ser otimizados os processos administrativos afim de cessar o uso de processos manuais, para centralizar e padronizar informações.

Com o uso de componentes da biblioteca PrimeNG e do servidor de aplicação NGINX foi possível criar um software robusto e seguro, arquitetado de maneira a permitir a futura expansão e manutenção. As funcionalidades requisitadas foram atendidas, o sistema conta com uma interface amigável e responsiva, que pode ser visualizada em qualquer dispositivo e em qualquer localidade, necessitando apenas de um navegador web e acesso a internet.

A praticidade e facilidade de uso foi pensada especialmente na adaptação dos usuários, que passarão do papel e ferramentas alternativas ao uso do sistema para unificação e melhoria dos processos. A segurança foi tratada em prioridade, pois este é um sistema que trata de dados sensíveis e precisa ser contemplado ferramentas de prevenção de acessos indesejados, agindo em conformidade com as normas de segurança e responsabilidade com o material humano gerenciado. Além dessas funcionalidades, no auxílio ao acesso rápido de relatórios foi adicionado um Dashboard para visualização dos dados atuais dos projetos e auxílio na tomada de decisão.

O projeto será doado a Banda e Orquestra Sinfônica Municipal de Bauru e irá reduzir a carga de trabalho burocrático dos professores e gestores, para permitir que estes se concentrem mais no ensino musical e no desenvolvimento artístico dos alunos. Projetos culturais musicais promovem o fortalecimento de vínculos, apresentam diversidade cultural e podem transformar diferentes contextos por meio da educação, disciplina, da expressão artística e criatividade. Acredita-se que otimizar a gestão acadêmica e administrativa, contribuirá para que os projetos continuem promovendo o acesso à cultura e impactando a vida de tantos jovens e crianças na cidade e região de Bauru.

Referências

- [1] (2024). *Banda Sinfônica Municipal*. Disponível em: https://www2.bauru.sp.gov.br/cultura/banda_municipal.aspx Acesso em: novembro de 2024.
- [2] (2024). *Emusys*. Disponível em: <https://emusys.com.br/#> Acesso em: novembro de 2024.
- [3] (2024). *PrimeNG*. Disponível em: <https://primeng.org/> Acesso em: dezembro de 2024.
- [4] (2024). *Sakai*. Disponível em: <https://sakai.primeng.org/> Acesso em: dezembro de 2024.
- [5] (2024). *Sismusica*. Disponível em: <https://www.sismusica.com.br/> Acesso em: novembro de 2024.
- [6] (2024). *Webmusica*. Disponível em: <https://sistemawebmusica.com.br/> Acesso em: novembro de 2024.
- [7] Jansen, P. (2024). *TIOBE Index for November 2024*. Disponível em: <https://www.tiobe.com/tiobe-index/> Acesso em: novembro de 2024.
- [8] Leah Kaufman, B. W. (1998). *Too Much of a Good Thing? Identifying and Resolving Bloat in the User Interface: A CHI 98 Workshop*.
- [9] Zandstra, M. (2016). *PHP Objects, Patterns, and Practice*. p. 157.