

SEPAIS: Ferramenta de comunicação escolar abrangente

Lucas Garzuze Cordeiro¹, Pietro Francisco Andrade Sotile¹,
Victor Hugo Rodrigues Macharete¹, Nereu Moreira da Silva Filho¹, Eduardo Tieppo¹

¹Instituto Federal do Paraná (IFPR) – Campus Pinhais.
Pinhais, Paraná, Brasil.

{lucasgarzuze, pietro.sotile, macharetevic}@gmail.com, {nereu.filho, eduardo.tieppo}@ifpr.edu.br

Resumo. A interlocução desempenha um papel vital na gestão e organização de qualquer instituição. Por conta disso, a falta de métodos e ferramentas que facilitem esse processo pode trazer efeitos indesejados. Principalmente, no que diz respeito a instituições de ensino, que possuem interações complexas e multifacetadas. Portanto, este trabalho propõe um sistema digital denominado SEPAIS, implementado no contexto do Instituto Federal do Paraná, projetado para melhorar a comunicação entre a Seção Pedagógica e de Assuntos Estudantis, os responsáveis pelos alunos e os servidores da portaria. Com a implementação bem-sucedida do SEPAIS, espera-se que a automatização dos processos reduza erros e sobrecarga de servidores.

Abstract. Communication is a critical element in the management and organization of any institution. The absence of effective methods and tools to support this process can lead to unintended consequences, particularly in educational institutions, where interactions are inherently complex. This study presents the development of a digital system, SEPAIS, implemented within the Instituto Federal do Paraná context. The primary objective of SEPAIS is to enhance communication between the Pedagogical and Student Affairs Section, student guardians, and school gatekeepers. The anticipated outcome of SEPAIS is the reduction of human errors and work overload through critical process automation.

1. Introdução

A comunicação faz parte da existência humana desde seu primórdio e sempre foi uma ferramenta essencial de sobrevivência. Nesse contexto, esse fator cultural possui um papel fundamental na organização de projetos e estratégias em organizações, como empresas e famílias. Portanto, para alcançar uma forma de se autoexpressar para outros de maneira eficiente, é necessário que os indivíduos usem modos de comunicação adequados [Genç 2017].

De forma análoga, a intercomunicação também desempenha um papel crucial na gestão de processos. Um gerente precisa ser um comunicador eficaz, e nenhuma organização pode ter sucesso, progredir ou construir reputação sem habilidades de comunicação eficazes [Radovic Markovic e Salamzadeh 2018]. Nesse sentido, possuir ferramentas adequadas para transmitir mensagens entre diferentes níveis organizacionais é vital para um bom funcionamento administrativo.

Essa necessidade se torna ainda mais evidente em instituições complexas, como hospitais, organizações não governamentais e instituições de ensino, onde a comunicação clara e eficiente é crucial para a coordenação de atividades, a tomada de decisões e a construção de uma cultura empresarial positiva. Reduzindo o escopo ao âmbito escolar, conforme objetiva este trabalho, a comunicação assertiva desempenha um papel fundamental no clima organizacional e está relacionada com a liderança adotada pelo diretor escolar e sua equipe [Soares et al. 2022]. Há como exemplo os Institutos Federais, que apresentam uma gama complexa e multifacetada de operações administrativas e pedagógicas que têm a interlocução como base.

Os Institutos Federais (IF) são instituições de ensino pluricurriculares espalhadas pelo Brasil em várias unidades. Essa rede de institutos é especializada no fornecimento de educação profissional técnica, de forma a integrar conhecimentos sobre tecnologias e métodos de uma área com os princípios pedagógicos adequados para cada uma das diversas modalidades de estudo oferecidas nas unidades das organizações de aprendizado [BRASIL 2008].

As instituições supracitadas possuem um setor especializado no auxílio dos processos de ensino-aprendizado entre os estudantes e os professores, a SEPAE (Seção Pedagógica e de Assuntos Estudantis). A equipe da SEPAE é constituída por servidores nos cargos de técnicos em assuntos educacionais, psicólogos e pedagogos que contribuem para o desenvolvimento dos alunos, principalmente na relação entre o ambiente escolar, familiar e a comunidade [IFPR 2021]. Nesse contexto, a comunicação com os responsáveis pelos alunos se dá em grande parte por meio de e-mail e WhatsApp institucionais, o que sobrecarrega os funcionários desse setor e aumenta a probabilidade de erros humanos devido à natureza manual do processo.

Diante do exposto, propõe-se um software denominado SEPAIS, projetado para melhorar a comunicação entre a Seção Pedagógica e de Assuntos Estudantis, os responsáveis dos alunos e os servidores da portaria, oferecendo funcionalidades específicas para cada grupo de usuários. Os servidores da SEPAE podem liberar antecipadamente os alunos, enviar recados digitais e visualizar o histórico de liberações. Já os responsáveis pelos alunos podem solicitar a liberação antecipada dos estudantes, receber solicitações de liberação e verificar os recados enviados pelos servidores. Por fim, os colaboradores da portaria, responsáveis pelo acesso à instituição, podem confirmar a saída dos alunos, resultando no envio de uma notificação aos celulares dos responsáveis informando a saída.

De maneira tangível, este trabalho apresenta o desenvolvimento do SEPAIS, um software que visa facilitar as interações entre a SEPAE e os responsáveis dos estudantes, auxiliando em processos como liberação antecipada de alunos e envio de recados.

Além disso, o projeto possui os seguintes objetivos específicos:

- i. Modelar um banco de dados para armazenamento de informações do sistema de comunicação digital;
- ii. Criar uma interface para auxílio na liberação dos alunos, caso sejam liberados antes do horário regular;
- iii. Elaborar uma função que notifique — em caso de liberação antecipada — o horário da saída dos estudantes para os seus responsáveis cadastrados;
- iv. Criar uma interface que permita aos responsáveis justificar a saída antecipada de seu(s) dependente(s).

Como mencionado anteriormente, a forma de intercomunicação utilizada muitas vezes limita o potencial da equipe pedagógica, causando sobrecargas e empecilhos tanto aos funcionários como aos estudantes do IF. Por exemplo, quando ocorre a liberação dos alunos antes do previsto pelo horário oficial de aulas, os discentes necessitam enviar uma mensagem aos seus responsáveis para que estes informem a SEPAE que a saída daquele estudante foi autorizada. Depois disso, a informação sobre a autorização deve ser direcionada à portaria do instituto por meio de uma mensagem via WhatsApp. Posteriormente, o educando deve se dirigir a portaria e informar seu nome, para que o porteiro possa verificar manualmente se o nome citado está presente na lista enviada pela equipe pedagógica para, enfim, liberá-lo. Em síntese, esse processo como um todo possui etapas que dependem obrigatoriamente de diversos agentes, assim aumentando o tempo empregado e a probabilidade de erros humanos na execução desse método.

Existem propostas que tentam mitigar os problemas gerados por essa abordagem administrativa. Essas soluções costumam envolver equipamentos adicionais, como catracas e acessos por biometria. Dentre eles, há o aplicativo *Biomessenger* [BioMessenger 2014], que, para manter os pais informados, envia automaticamente notificações com a data e hora que o aluno registra sua impressão digital na catraca, informando a entrada ou saída das dependências escolares. Outro software é o *Edu Go!* [AgendaEdu 2024], que permite ao responsável avisar com antecedência sobre sua chegada à escola, possibilitando, assim, maior agilidade no processo de liberação do aluno.

Entretanto, apesar de possuírem pontos fortes dentro de seus contextos, as soluções citadas acima apresentam lacunas que interferem em aspectos fundamentais da comunicação na gestão escolar e que podem resultar em perdas no processo de controle acadêmico realizado pela Seção Pedagógica e de Assuntos Estudantis. Como exemplo, a solução *Biomessenger* depende exclusivamente de catracas. Isso é problemático pois, em um contexto público, é pouco factível a compra de catracas em larga escala para várias instituições, pois cada uma delas possui necessidades e orçamentos distintos. Ademais, o software *Edu Go!* apenas permite que o responsável avise com antecedência que está chegando à escola. Essa funcionalidade não contribui para a solução dos problemas que a Seção Pedagógica e de Assuntos Estudantis enfrenta atualmente, ao passo que o que ocasiona a sobrecarga dos servidores é ter que se comunicar com os pais manualmente quando os educandos são liberados antecipadamente.

Nesse sentido, o software aqui proposto objetiva contribuir para a solução de tais pendências com a aplicação de um sistema que não dependa de dispositivos de hardware específicos, como catracas, para funcionar. O SEPAIS necessita somente de computadores com acesso à Internet para os servidores e portaria, e um dispositivo móvel com acesso à Internet para os responsáveis. As funcionalidades de cada ator no processo de comunicação escolar forammeticulosamente projetadas para a integração e otimização da interação entre as partes, promovendo uma gestão mais ágil e eficaz.

Os servidores da SEPAE teriam a capacidade de solicitar autorização para a saída antecipada dos alunos, enviar recados e acessar o histórico de liberações. Os responsáveis pelos alunos poderiam confirmar e solicitar a liberação antecipada e verificar as mensagens enviadas pelos funcionários. E os funcionários da portaria poderiam confirmar a saída dos alunos, resultando no envio de uma notificação aos dispositivos móveis dos responsáveis, informando o horário de saída.

A automação de processos — como a liberação de alunos antes do horário regular e a justificativa de saídas antecipadas — poderá reduzir a sobrecarga de trabalho, minimizar erros e proporcionar uma comunicação mais eficaz. Além disso, a funcionalidade de notificação manterá os responsáveis informados sobre a situação de seus dependentes. Em última análise, o sistema se propõe a modernizar a comunicação no IFPR, otimizando processos, reduzindo falhas e adaptando-se às necessidades dos usuários. Preparando, dessa forma, a instituição para os desafios da comunicação na era digital.

2. Método

Este trabalho visa facilitar as interações entre a Seção Pedagógica e de Assuntos Estudantis e os responsáveis dos estudantes, especificamente, no que diz respeito a processos como liberação antecipada de alunos e envio de recados. Para que esse objetivo possa ser alcançado, o sistema é composto de três tipos de usuários: servidor da SEPAE, funcionário da portaria e responsável do educando. Cada usuário possui permissões e funcionalidades específicas, que se integram e interagem construindo, assim, a arquitetura do sistema. A Figura 1 exemplifica o esquema geral do software.

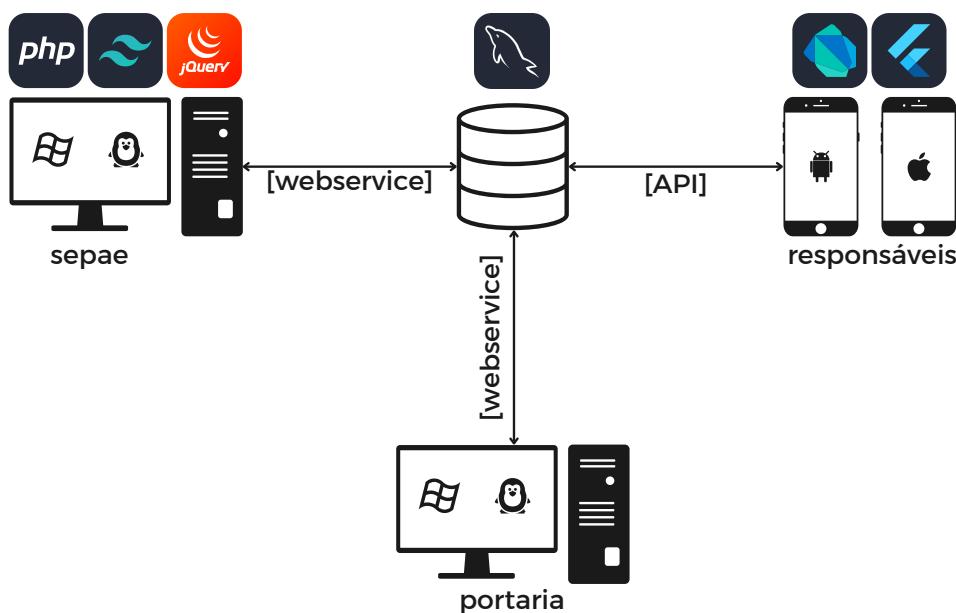


Figura 1. Arquitetura da aplicação

A ferramenta utilizada pelos servidores da SEPAE e da portaria será executada em um navegador, enquanto os responsáveis utilizarão um aplicativo para dispositivos móveis. Para a estruturação e estilização da primeira foram utilizados HTML e CSS em conjunto com Tailwind [Wathan et al. 2024], uma biblioteca CSS baseada em classes. Além disso, para realizar a conexão com o banco de dados a linguagem PHP foi utilizada, em conjunto com a biblioteca jQuery, que, através de AJAX, requisita informações do banco de dados por meio de um *Web Service*. O banco de dados foi construído a partir da linguagem estruturada SQL, e gerenciado com o sistema de gerenciamento de banco de dados MySQL [Axmark et al. 2024].

Por fim, o aplicativo móvel é desenvolvido com Flutter, um *framework* da linguagem Dart. As interações do aplicativo com o banco de dados são mediadas por uma *API* (*Application Programming Interface*, ou Interface de Programação de Aplicação, em português) e autenticadas pela ferramenta JWT (JSON Web Token) [Jones et al. 2015]. Cada parte que integra o sistema será abordada de forma detalhada nas seções seguintes.

2.1. Modelagem de dados

O banco de dados é responsável por armazenar os dados necessários ao funcionamento do sistema, como informações sobre alunos, responsáveis, recados para os responsáveis, liberações pela SEPAE e pelos responsáveis, servidores da SEPAE e da portaria. Esses dados organizados de forma relacional são gerenciados por meio da linguagem de consulta estruturada SQL e do sistema de gerenciamento de banco de dados (SGBD) MySQL [Axmark et al. 2024].

No contexto do SEPAIS, o banco de dados do sistema pode ser descrito pelo Diagrama Entidade-Relacionamento (DER) exibido na Figura 2.

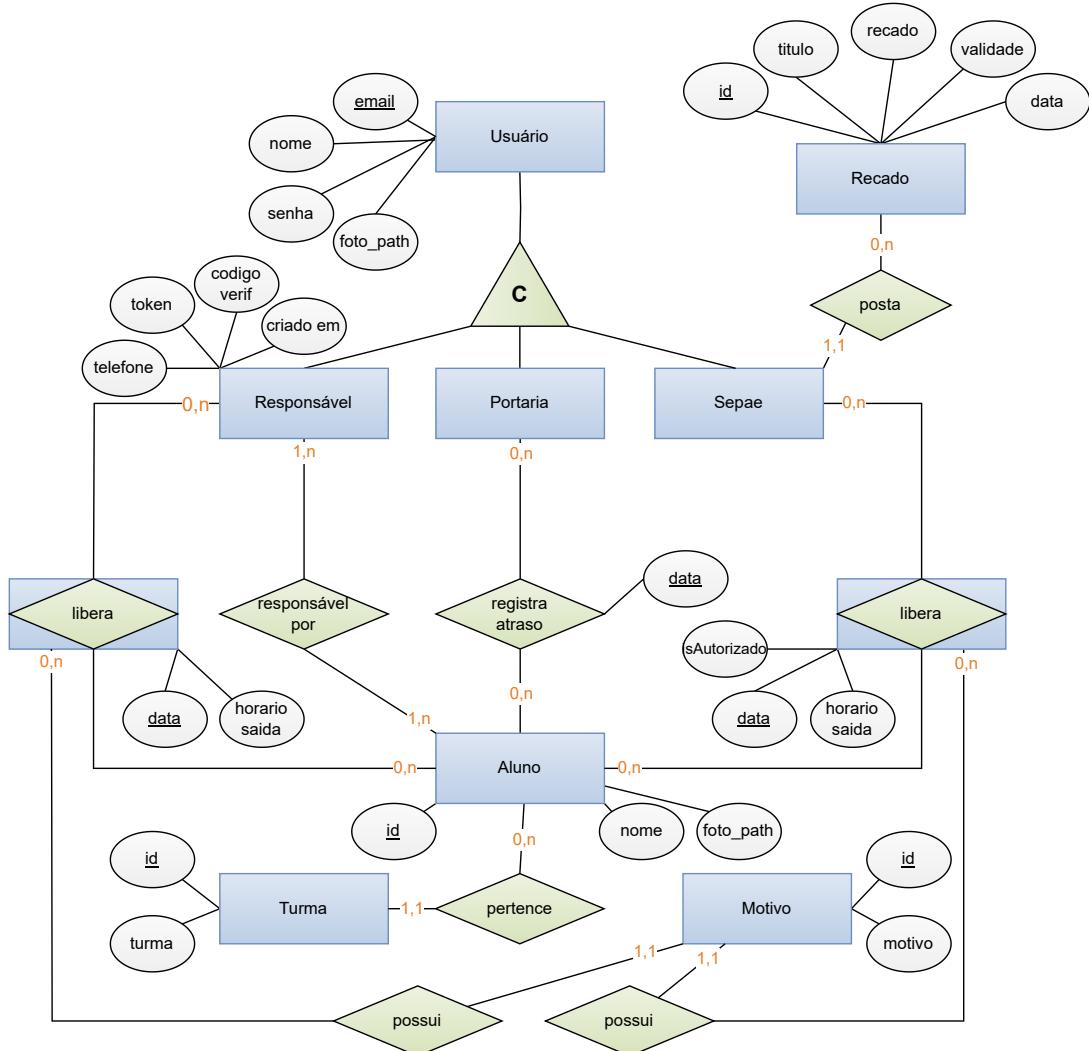


Figura 2. Diagrama Entidade Relacionamento da aplicação

As entidades expressas pelo Diagrama Entidade Relacionamento (Figura 2) são Usuário — que se especializa de forma mutualmente exclusiva em Responsável, Portaria e Sepae —, Recado, Liberação por responsável, Liberação pela Sepae, Aluno, Turma e Motivo. As entidades especializadas são mutualmente exclusivas pelo motivo do sistema não permitir a existência de um usuário que desempenha a função de mais de um ator. Por exemplo, caso um servidor da SEPAE tenha um dependente no instituto, ele não poderá usar o aplicativo com o seu e-mail institucional, que é utilizado no website da SEPAE. Por se tratar de uma relação mais particular, o e-mail pessoal do servidor seria o ideal para o uso do aplicativo.

As entidades supracitadas presentes no diagrama se relacionam da seguinte maneira:

- O relacionamento entre Aluno e Turma possui cardinalidade 0,n : 1,1;
- O relacionamento entre Aluno e Responsável (responsável por) possui cardinalidade 1,n : 1,n;
- O relacionamento entre Aluno e Responsável (libera) possui cardinalidade 0,n : 0,n;
- O relacionamento entre Aluno e Sepae possui cardinalidade 0,n : 0,n;
- O relacionamento entre Aluno e Portaria possui cardinalidade 0,n : 0,n;
- O relacionamento entre ambas Liberação e Motivo possui cardinalidade 0,n : 1,1;
- O relacionamento entre Sepae e Recado possui cardinalidade 1,1 : 0,n.

Ambas as entidades Liberação por responsável e Liberação pela Sepae participam de agregações com uma relação de liberação. No entanto, a Liberação pela Sepae se diferencia por possuir o atributo isAutorizado, que indica se a saída do aluno foi autorizada por um responsável. Esse atributo é essencial para verificar se o aluno pode ou não sair de maneira antecipada do instituto, pois a saída de um estudante do instituto sem a autorização dos responsáveis pode causar impecilhos legais ao instituto e aos servidores que realizaram a liberação.

O Instituto Federal do Paraná intercala as turmas de alguns cursos de graduação, de modo a evitar sobrecarga de professores e insuficiência de salas de aulas para todas as turmas. Devido a isso, a relação entre Aluno e Turma permite que existam turmas que não possuem nenhum aluno em determinados períodos do ano.

No contexto de realização das funções do sistema, as informações contidas no banco de dados podem ser resgatadas através de consultas armazenadas no próprio SGBD por meio de visões, ou consultas externas ao banco de dados, ambas opções podem utilizar as chaves primárias das entidades como parâmetros de consulta.

2.2. Interface Web de controle para SEPAE e Portaria

A plataforma web foi concebida com o propósito de centralizar e otimizar as operações da Seção Pedagógica e da Portaria. Os servidores pedagógicos podem utilizar a ferramenta para realizar diversas tarefas, como autorização de saídas de alunos, elaboração de comunicados e consulta aos históricos de atrasos e liberações. Por sua vez, os funcionários da portaria têm acesso a uma interface que permite confirmar a saída de alunos, registrar horários de saída e documentar atrasos.

O sistema foi desenvolvido utilizando uma arquitetura de três camadas (apresentação, lógica de negócios e dados) [IBM 2024], com a seguinte estruturação:

- i. Camada de apresentação: construída com HTML, Tailwind CSS, JavaScript e jQuery, tendo como principal função exibir a informações e permitir a interação do usuário com o sistema;
- ii. Camada de lógica de negócios: implementada na linguagem *server-side* PHP, sendo responsável por processar as requisições de um usuário, acessar o banco de dados e gerar respostas a requisições;
- iii. Camada de dados: Baseada em MySQL, armazenando as informações sobre alunos, liberações, atrasos e outras entidades do sistema.

A Figura 3 exemplifica de forma gráfica quais são as funcionalidades que devem ser acessadas pelos funcionários da SEPAE a partir da interface gráfica.

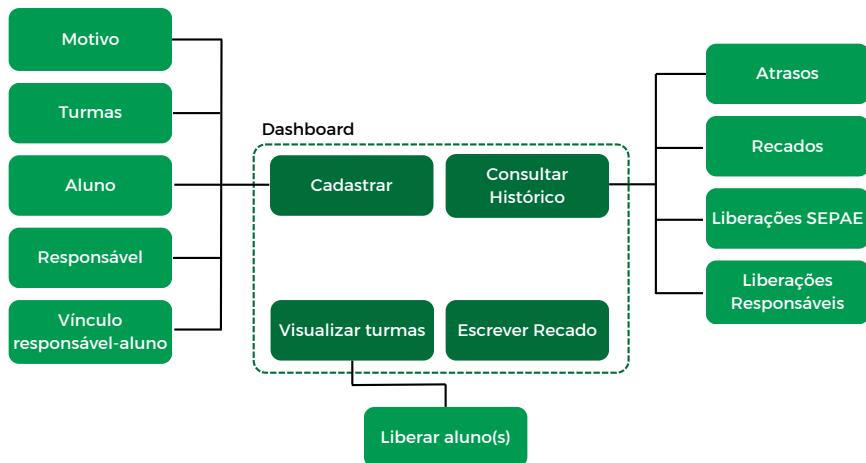


Figura 3. Diagrama de funcionalidades

Ao acessar o sistema web, os usuários terão à sua disponibilidade um painel com várias funções, necessárias para a integração de diversas tarefas administrativas. Separando categoricamente, o servidor da SEPAE pode cadastrar: motivos de liberação antecipada; turmas novas; alunos de determinadas turmas, responsáveis de alunos e a associação entre eles.

Adicionalmente, os usuários também podem consultar históricos de atrasos dos alunos; recados enviados pela SEPAE; liberações realizadas pela SEPAE ou pelos responsáveis. Além de visualizar a relação de todos os alunos de uma turma, com a possibilidade de selecionar determinados educandos para liberar antecipadamente, e poder escrever recados que serão enviados para os responsáveis. É imperativo que todas as funcionalidades internas do sistema sejam acessadas apenas por usuários autenticados, a fim de garantir que nenhum agente externo possa enviar dados maliciosos ao sistema. Para essa finalidade, um sistema de credenciamento por login é suficiente para garantir a integridade das interações.

2.3. Web Service/API

A organização das informações essenciais para o sistema foi realizada através de um *Web Service*, que permite que dados do banco de dados sejam acessados por meio de requisições HTTP e exibidos no formato JSON. O *Web Service* é utilizado pela aplicação para requisitar as informações necessárias para o funcionamento desta. As consultas podem ser de leitura (*read*), de inserção (*insert*) ou de atualização (*update*) e podem necessitar de determinados parâmetros.

Os arquivos do *Web Service* foram desenvolvidos na linguagem PHP, de forma a realizar os comandos para operações no banco de dados através desses arquivos. Nas aplicações Web, os arquivos do *Web Service* são acessados por requisições AJAX por meio da biblioteca JQuery, da linguagem JavaScript. Essas requisições permitem a passagem de parâmetros, que são associados às consultas dos arquivos PHP para manipular dados de forma mais precisa.

No contexto do *Web Service*, as consultas mais notáveis para o funcionamento do sistema Web e seus parâmetros estão expressos abaixo:

- Leitura de alunos por turma, que recebe a turma como parâmetro;
- Leitura de recados, que recebe data atual como parâmetro;
- Leitura de alunos liberados, que recebe data atual como parâmetro;
- Inserção de alunos atrasados, que recebe data, e-mail do funcionário da portaria e id do aluno como parâmetros;
- Inserção de recados, que recebe data, validade, título, conteúdo e e-mail do funcionário da SEPAE como parâmetros;
- Inserção de liberações pela SEPAE, que recebe data, e-mail do funcionário da SEPAE, id do aluno e motivo como parâmetros;
- Atualização de liberações de alunos, que recebe data, horário de saída e id do aluno como parâmetros.

Além disso, a partir das consultas feitas para o *Web Service* — que foram transformadas em funções PHP —, foi criada uma *API*. Essa *API* objetiva mediar as interações entre o aplicativo utilizado pelos responsáveis e o banco de dados. Todas as requisições são baseadas no sistema de credenciais JWT, ou JSON Web Token, um padrão aberto, que estabelece uma maneira compacta e independente de transmitir dados com segurança entre partes como um objeto JSON [Jones et al. 2015]. Essa implementação é necessária para garantir que apenas usuários autenticados possam requisitar dados para a *API*.

A autenticação dos responsáveis funciona da seguinte forma: ao usuário enviar credenciais de login válidas, a *API* retorna ao cliente um *token* (chave de acesso) de autenticação, responsável pela validação de requisições feitas pelo cliente. Esse *token* é armazenado localmente pelo usuário, e deve ser incluído em todas as requisições subsequentes.

Dessa forma, o servidor recebe o *token*, verifica sua integridade e recupera dados sobre o usuário a partir dele. Estes dados são utilizados como parâmetros para as consultas requisitadas pelo cliente. Por exemplo, para requisitar os dependentes de determinado responsável, é preciso ter como parâmetro o e-mail desse responsável. A Figura 4 exemplifica de maneira simplificada esse processo.

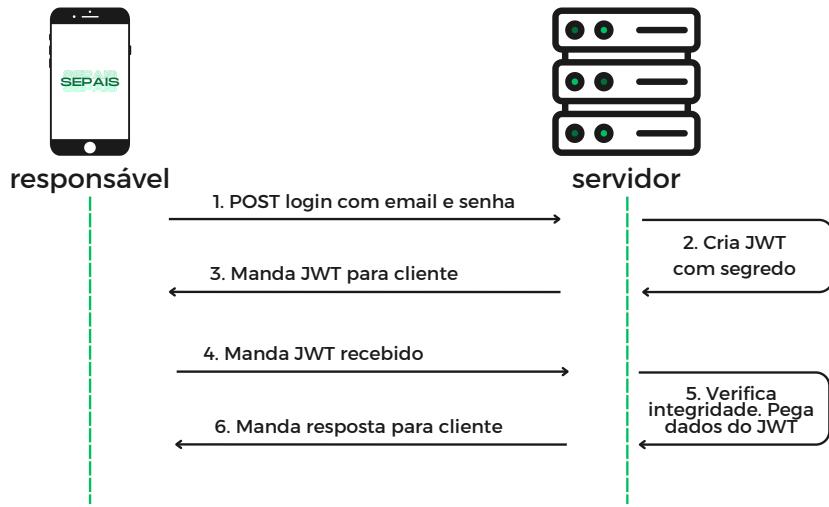


Figura 4. Autenticação via JWT

Além da consulta necessária para validar as credenciais do usuário, outras foram construídas para o funcionamento do aplicativo. Dentre elas, pode-se destacar:

- Requisitar dependentes de um responsável, tendo seu e-mail pessoal como parâmetro;
- Requisitar dependentes do responsável que foram liberados pela SEPAE, tendo seu e-mail como parâmetro;
- Autorizar a saída de um dependente que foi liberado pela SEPAE, tendo id do aluno como parâmetro;
- Requisitar dependentes autorizados a sair, tendo e-mail do responsável como parâmetro;
- Requisitar horário de saída dos alunos da instituição, tendo e-mail do responsável como parâmetros;
- Liberar dependentes, que utiliza e-mail do responsável, id do aluno liberado e motivo da liberação como parâmetros;
- Requisitar recados enviados pela Seção Pedagógica.

2.4. Funções via aplicativo para Responsáveis

O aplicativo foi desenvolvido com o objetivo de concentrar as ações que podem ser tomadas pelos responsáveis. Tais quais: autorizar a saída de seu dependente, caso este tenha sido liberado mais cedo; liberar seu dependente, caso precise que ele saia antecipadamente, anexando uma justificativa para a liberação; e ter acesso a recados enviados pela Seção Pedagógica. Para alcançar esses objetivos, o aplicativo móvel foi desenvolvido com o *Framework Flutter*, da linguagem de programação Dart.

Outro aspecto na utilização do aplicativo é o cadastro dos responsáveis, necessário para o primeiro acesso ao aplicativo. Esse processo funcionará da seguinte forma:

1. A SEPAE cadastrará o responsável, com seu nome completo e e-mail, e o vínculo que ele tem com determinado estudante;
2. O responsável, por sua vez, entrará no aplicativo utilizando o mesmo e-mail cadastrado pela SEPAE;
3. Um e-mail com um código de verificação será enviado para o endereço eletrônico do responsável;
4. Ao digitar o código enviado no aplicativo, o responsável poderá criar sua senha de acesso ao aplicativo.

Esse método garante que quem está criando a conta realmente seja o responsável pelo estudante. Esta etapa diferencia o processo proposto pelo SEPAIS do processo anterior em utilização — que não possuía nenhuma forma eficaz de verificar se quem liberava um aluno era, de fato, responsável por aquele aluno —, ao passo que adiciona uma camada de segurança no processo de liberação.

Ademais, outra funcionalidade relevante para o aplicativo móvel são as notificações *push* (notificações enviadas em tempo real pelo aplicativo). Quando a Seção Pedagógica liberar alunos, os responsáveis destes receberão uma notificação informando o ocorrido. Então, ao clicar na notificação, os responsáveis poderão escolher se autorizam ou não a saída do aluno. Se sim, eles receberão outra notificação, informando quando o aluno saiu assim que ele passar pela portaria e o funcionário registrar sua saída.

A Figura 5 exemplifica de forma gráfica quais são as funcionalidades que podem ser acessadas pelos responsáveis a partir do aplicativo.

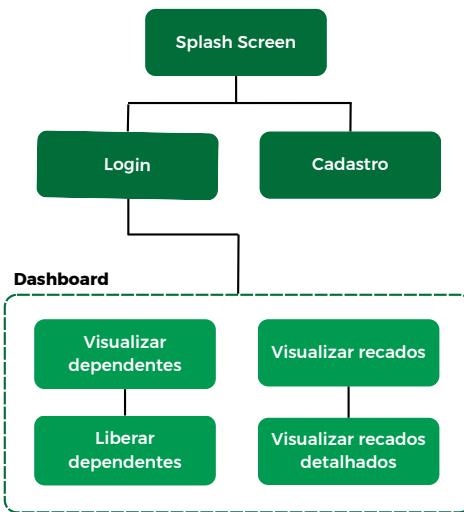


Figura 5. Diagrama de Funcionalidades do Aplicativo

Ao acessar a aplicação, os responsáveis irão se deparar com uma tela de apresentação, que os possibilita se cadastrar — ou seja, definir sua senha, conforme explicado anteriormente —, caso seja a primeira vez que utilizam o sistema, ou fazer login. Após o login bem-sucedido, os usuários têm acesso a uma tela com a relação de seus dependentes, com a possibilidade de liberá-los antecipadamente, anexando um motivo. Além de poder visualizar recados enviados pela Seção Pedagógica, com acesso a uma tela de detalhe, que mostra a mensagem completa, caso não caiba na lista de recados.

3. Resultados

As subseções a seguir descrevem os resultados obtidos a partir dos métodos descritos na Seção 2. Cada subseção apresenta uma combinação de elementos como tabelas, figuras e informações técnicas para descrever de maneira detalhada os resultados dos respectivos métodos empregados.

3.1. Banco de dados

O banco de dados do sistema SEPAIS foi modelado de acordo com o DER apresentado na Figura 2. A partir das relações e entidades presentes nesse diagrama, foram criadas 12 tabelas, expressas pelos dicionários de dados representados nas Tabelas 3-12.

O formato padrão dos dados de cada coluna das tabelas do sistema é definido pelo tipo de dado designado ao campo. Por exemplo, um campo designado como DATETIME somente aceita valores no formato ‘YYYY-MM-DD HH:MM:SS’. Desse modo, qualquer inserção feita no banco de dados com valores inválidos não será realizada e retornará um erro.

O campo `isAutorizado` apresentado na Tabela 8 apenas aceita os valores 0 ou 1, que representam, respectivamente, as condições falso ou verdadeiro. Esse campo tem como valor padrão 0, ou seja, será falso quando uma liberação pela SEPAE for inserida no banco de dados sem preencher esse campo. Desse modo, o estudante não está autorizado a sair no exato momento que é liberado pela SEPAE e só passa a ter sua liberação autorizada quando esse campo é atualizado para verdadeiro, quando o responsável autoriza a saída.

A coluna `senha` presente na Tabela 4 aceita valores nulos. Entretanto, os usuários do sistema apenas possuem senha nula em casos específicos e temporários. Por exemplo, somente responsáveis que nunca fizeram login no aplicativo podem possuir senha nula, pois o cadastro de responsáveis é feito pela SEPAE e neste momento a senha não é definida. Porém, quando o responsável for fazer login pela primeira vez, poderá definir sua senha, atualizando, assim, a informação no banco de dados.

Na tabela 5, as colunas `codigo_verif` e `criado_em` servem para armazenar o código de verificação — que é enviado no e-mail do responsável quando ele vai criar sua senha — e o horário em que foi criado. A persistência desses dados é importante para aferir se o código digitado pelo responsável é válido, ou seja, se é igual ao criado e se não excedeu o tempo limite de 15 minutos após sua criação. Além disso, a coluna `token` serve para armazenar o *token* de identificação do dispositivo móvel do responsável.

Tabela 1. Dicionário de Dados (Tabela: turma)

Coluna	Tipo de dado	Restrições	Exemplo de instância
<code>id</code>	INT	PK, AI	1
<code>turma</code>	VARCHAR(5)	NN	‘INFO4’

Tabela 2. Dicionário de Dados (Tabela: sepae)

Coluna	Tipo de dado	Restrições	Exemplo de instância
<code>email</code>	VARCHAR(144)	PK, FK	‘nereu@ifpr.edu.br’

Tabela 3. Dicionário de Dados (Tabela: aluno)

Coluna	Tipo de dado	Restrições	Exemplo de instância
id	INT	PK, AI	5
nome	VARCHAR(144)	NN	'Bart Simpson'
foto_path	VARCHAR(144)	NN	'static/bart.png'
turma_id	INT	FK, NN	1

Tabela 4. Dicionário de Dados (Tabela: usuário)

Coluna	Tipo de dado	Restrições	Exemplo de instância
email	VARCHAR(144)	PK	'homer@gmail.com'
nome	VARCHAR(144)	NN	'Homer Simpson'
senha	VARCHAR(144)		'10SoQnabBBJN....'
foto_path	VARCHAR(144)	NN	'static/homer.jpg'

Tabela 5. Dicionário de Dados (Tabela: responsavel)

Coluna	Tipo de dado	Restrições	Exemplo de instância
email	VARCHAR(144)	PK, FK	'homer@gmail.com'
telefone	VARCHAR(11)	NN	'11321312313'
token	VARCHAR(200)		'f8oTZL34Ss-....'
codigo_verif	VARCHAR(6)		'154945'
criado_em	TIMESTAMP		'2024-12-04 11:46:22'

Tabela 6. Dicionário de Dados (Tabela: portaria)

Coluna	Tipo de dado	Restrições	Exemplo de instância
email	VARCHAR(144)	PK, FK	'cerberus@ifpr.edu.br'

Tabela 7. Dicionário de Dados (Tabela: aluno_atrasado)

Coluna	Tipo de dado	Restrições	Exemplo de instância
aluno_id	INT	PK, FK	5
data	DATETIME	PK	'2024-09-21 12:28:33'
portaria_email	VARCHAR(144)	FK, NN	'cerberus@ifpr.edu.br'

Tabela 8. Dicionário de Dados (Tabela: sepae_libera_aluno)

Coluna	Tipo de dado	Restrições	Exemplo de instância
aluno_id	INT	PK, FK	5
data	DATETIME	PK	'2024-09-21 12:28:33'
saida	DATETIME		'2024-09-21 12:32:36'
motivo_id	INT	FK, NN	4
sepae_email	VARCHAR(144)	PK, FK	'nereu@ifpr.edu.br'
isAutorizado	TINYINT(1)		1

Tabela 9. Dicionário de Dados (Tabela: responsavel_has_aluno)

Coluna	Tipo de dado	Restrições	Exemplo de instância
aluno_id	INT	PK, FK	5
responsavel_email	VARCHAR(144)	PK, FK	'homer@gmail.com'

Tabela 10. Dicionário de Dados (Tabela: responsavel_libera_aluno)

Coluna	Tipo de dado	Restrições	Exemplo de instância
aluno_id	INT	PK, FK	5
data	DATETIME	PK	'2024-09-21 12:28:33'
saida	DATETIME		'2024-09-21 12:32:36'
motivo_id	INT	FK, NN	4
responsavel_email	VARCHAR(144)	PK, FK	'homer@gmail.com'

Tabela 11. Dicionário de Dados (Tabela: motivo)

Coluna	Tipo de dado	Restrições	Exemplo de instância
id	INT	PK, AI	4
motivo	VARCHAR(45)	NN	'Mal-estar'

Tabela 12. Dicionário de Dados (Tabela: recado)

Coluna	Tipo de dado	Restrições	Exemplo de instância
id	INT	PK, AI	10
titulo	VARCHAR(144)	NN	'Reunião do pais'
recado	VARCHAR(1000)	NN	'A reunião vai...'
data	DATE	NN	'2024-09-21'
validade	DATE		'2024-09-30'
sepaes_email	VARCHAR(144)	FK, NN	'nereu@ifpr.edu.br'

3.2. SEPAIS Web

Inicialmente, ao acessar a interface web, o usuário (servidor da SEPAE ou portaria) deverá fazer login com suas credenciais. Após o login bem-sucedido, os usuários têm acesso a um *Dashboard*. No caso dos pedagogos, eles podem escolher entre enviar um recado, requisitar históricos de liberações, atrasos e recados — que podem ser exportados para PDF e CSV — e cadastrar alunos, responsáveis, o vínculo entre eles, motivos de liberação e turmas. Já a portaria pode verificar os alunos que estão autorizados a sair. Para ambos usuários é possível visualizar uma relação de todas as turmas registradas, que, ao serem clicadas, exibem uma tabela com os alunos da turma e seus responsáveis. Para a SEPAE, eles podem selecionar os alunos que desejam liberar, enquanto para a portaria, eles podem registrar atrasos para determinados discentes.

A implementação da funcionalidade de login e autenticação se baseia no mecanismo de sessões. Sessões são áreas de memória reservadas para armazenar informações específicas de cada usuário durante a sua navegação no sistema, o que permite a manutenção do estado da aplicação entre diferentes páginas. Diferentemente dos *cookies*, essas informações não são persistidas no dispositivo do usuário após ele fechar o navegador, assim encerrando a sessão.

No momento da autenticação, duas variáveis de sessão são criadas: uma contendo o endereço de e-mail do usuário e outra armazenando um *hash* da senha — e não a senha em texto plano, por motivos de segurança. Essa estratégia permite identificar o usuário autenticado e restringir o acesso às funcionalidades internas do sistema apenas aos usuários devidamente logados.

A verificação da sessão ativa é centralizada no arquivo `secure.php`, que possui a função `secure_page()`. Esta verifica a existência das variáveis de sessão correspondentes ao e-mail e à senha. Caso alguma delas esteja ausente, a sessão é encerrada e o usuário é redirecionado para a página de login.

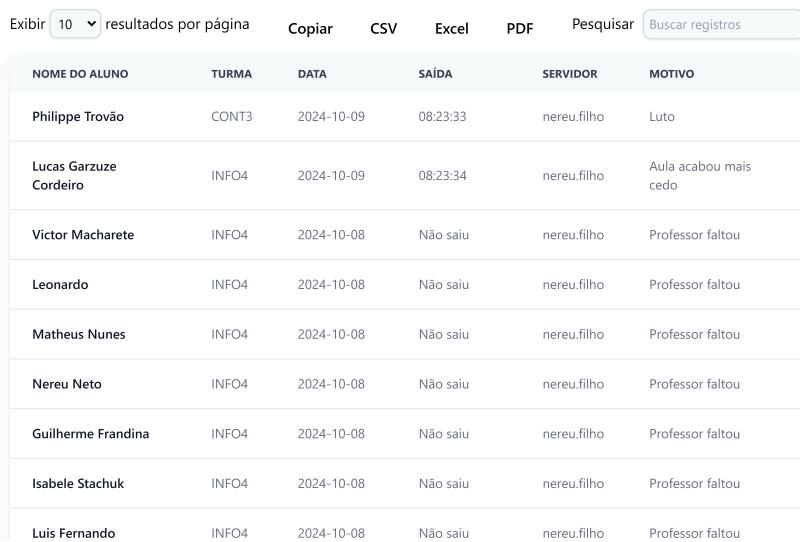


Figura 6. Dashboard

A página de *Dashboard* do sistema, expressa na Figura 6, possui carregamento dinâmico, em que cada funcionalidade é carregada sob demanda, sem a necessidade de recarregar toda a página ou abrir outro arquivo. Essa abordagem proporciona uma experiência de usuário mais fluida, uma vez que as diferentes funcionalidades são exibidas dentro da mesma página, em resposta a interações do usuário, como cliques em botões. Além da navegação por meio dos botões do *Dashboard*, o sistema também suporta o acesso direto a funcionalidades específicas através da URL. Ao clicar em um botão ou ao digitar diretamente no navegador um endereço com o formato `sepais.com/#nome-da-funcionalidade`, o sistema direciona o usuário para a funcionalidade desejada.

A interação do usuário com o sistema é gerenciada por eventos de clique associados a cada botão. Ao clicar em um botão, uma função JavaScript específica é acionada. Essa função, por sua vez, utiliza requisições AJAX para carregar dinamicamente o conteúdo necessário dentro de uma `div` pré-definida na página. Por exemplo, ao clicar no botão “Cadastrar Responsável”, a função `loadCadastrarResponsavel()` é executada, atualizando a interface com o formulário de cadastro de responsáveis. Essa abordagem permite uma experiência de usuário mais fluida, pois evita recargas completas da página.

Na visualização dos dados recuperados do *backend* em formato de tabelas, a biblioteca DataTables foi utilizada. Com ela é possível implementar um filtro de linhas, com a possibilidade de limitar o número de resultados, ou filtrar dados em qualquer coluna da tabela por meio de uma pesquisa em uma caixa de texto. Além disso, ela também permite exportar a tabela visualizada em arquivos PDF, XLSX ou CSV [SpryMedia 2024]. A Figura 7 exprime a visualização do histórico de liberações realizadas pela SEPAE.



The screenshot shows a table titled "Visualização do histórico de liberações pela SEPAE". At the top, there are buttons for "Exibir 10 resultados por página", "Copiar", "CSV", "Excel", "PDF", "Pesquisar", and a search input field "Buscar registros". The table has columns: NOME DO ALUNO, TURMA, DATA, SAÍDA, SERVIDOR, and MOTIVO. The data is as follows:

NOME DO ALUNO	TURMA	DATA	SAÍDA	SERVIDOR	MOTIVO
Philippe Trovão	CONT3	2024-10-09	08:23:33	nereu.filho	Luto
Lucas Garuze Cordeiro	INFO4	2024-10-09	08:23:34	nereu.filho	Aula acabou mais cedo
Victor Macharete	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou
Leonardo	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou
Matheus Nunes	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou
Nereu Neto	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou
Guilherme Frandina	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou
Isabele Stachuk	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou
Luis Fernando	INFO4	2024-10-08	Não saiu	nereu.filho	Professor faltou

Figura 7. Visualização do histórico de liberações pela SEPAE

O sistema utiliza modais para solicitar a confirmação do usuário em ações significativas. Um modal é um elemento de uma página web que aparece em frente ao conteúdo principal e move o sistema para um modo especial que requer interação do usuário. Eles geralmente são utilizados para direcionar a atenção do usuário para uma ação importante [Juviler 2022]. Um exemplo da implementação desse recurso é quando, após selecionar os estudantes para liberação, o servidor clica em Liberar, um modal aparece solicitando que o usuário selecione o motivo da liberação, para que então possa liberar os alunos, conforme expresso na Figura 8 (2).

A funcionalidade de validações interativas e *feedback* instantâneo em formulários utiliza expressões regulares para verificar a validade de entradas como e-mails e telefone em tempo real. As validações são realizadas após o usuário inserir dados, sem a necessidade de submeter o formulário. Essa abordagem melhora a usabilidade, pois previne erros e fornece ao usuário *feedback* através de mensagens de erro, como as implementadas com *SnackBar*s (mensagens de *feedback* temporárias) no código. Além de melhorar a experiência do usuário, essas validações reduzem a chance de submissões incorretas, pois otimizam a integridade dos dados e minimizam a necessidade de revisões futuras.

Ao selecionar os alunos para liberação, há a possibilidade de selecionar alguns alunos individualmente ou todos de uma vez, conforme expresso na Figura 8 (1). Essa funcionalidade envolve a manipulação do DOM (Document Object Model) e o gerenciamento de eventos. Utilizando jQuery é possível capturar eventos de clique no *checkbox* principal e propagar essa ação para os *checkboxes* subordinados, marcando ou desmarcando todos simultaneamente.

Essa interação ocorre por meio da iteração sobre os elementos filhos no DOM, utilizando o método `.each()`, que altera estado do *checkbox* subordinado de acordo com o estado do *checkbox* principal. Além disso, o estado de sincronização inversa, onde os *checkboxes* individuais atualizam o *checkbox* principal, assegura que a interface reflita as ações do usuário de forma consistente.

Na seção de cadastrar o vínculo entre os responsáveis e os alunos há o recurso de sugestão de resultados presentes no banco de dados, de forma dinâmica, a partir do que o usuário digita nas caixas de texto, conforme ilustrado na figura 8 (3). Isso é feito por meio das funções JavaScript `sugerirAlunos()` e `sugerirEmails()`, que fazem uma consulta no banco de dados utilizando o que o usuário digitou como parâmetro para buscar registros parecidos. Isso facilita o cadastro do vínculo responsável-aluno, pois a sugestão de resultados acelera o processo de achar o responsável correspondente ao aluno.

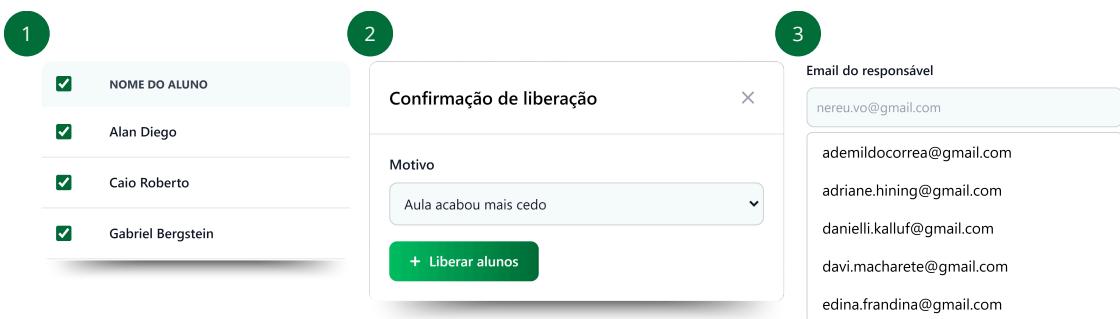


Figura 8. Exemplos de funcionalidades web

A realização de cadastros do sistema pode ser realizada de maneira unificada, com menos interferência humana, por meio da importação de dados presentes em um arquivo XLSX. Nesse caso, a leitura do arquivo é realizada por meio da biblioteca JavaScript `SheetJS`, que armazena os valores, agrupados por linha e colunas, em formato JSON. Após a leitura do documento, os campos relevantes ao sistema são filtrados e enviados como parâmetros de uma requisição AJAX para um arquivo PHP.

O arquivo `recebe_xlsx.php` recebe, valida e prepara os parâmetros da requisição AJAX de importação do arquivo XLXS para a inserção no banco de dados. De modo a realizar sequencialmente as seguintes operações para cada linha da planilha: inserir ou atualizar, caso já existentes, turmas, alunos, responsáveis e os vínculos entre eles. No contexto do IFPR, uma planilha com os dados citados anteriormente é gerada anualmente por meio de um formulário de matrícula obrigatório e a funcionalidade de importação foi projetada com base nesse arquivo XLSX.

A atualização da situação dos estudantes é um fator crítico para o funcionamento e a sustentabilidade do sistema ao decorrer do tempo, pois indivíduos que não possuem mais relação com a organização não devem participar das operações do sistema. Nesse contexto, a figura 9 ilustra uma funcionalidade que permite a atualização da condição dos alunos para aprovados ou reprovados.

Confirmar aprovações de alunos de ADM4		Confirmar desistências de alunos de ADM4	
<input type="checkbox"/>	NOME DO ALUNO	<input type="checkbox"/>	NOME DO RESPONSÁVEL
<input type="checkbox"/>	Groundskeeper Willie	<input type="checkbox"/>	Peppermint Patty
<input type="checkbox"/>	Lara Croft	<input type="checkbox"/>	Foxy Pirate, Schroeder
<input type="checkbox"/>	Mariana	<input type="checkbox"/>	Chica Chicken, Dr. Eggman
<input type="checkbox"/>	Ned Flanders	<input type="checkbox"/>	Dr. Eggman, Franklin

Figura 9. Funcionalidade para confirmar aprovação ou desistência

Caso a importação de dados por XLSX for realizada periodicamente, os alunos aprovados irão avançar para próxima turma automaticamente. Entretanto, caso o aluno desista do curso ou se forme, será necessário o uso da funcionalidade de atualização.

Todas as interações da interface web com o banco de dados são mediadas a partir de *Web Services*, que são acionados por meio de ferramentas AJAX. Por exemplo, para exibir alunos de uma determinada turma, é feita uma requisição AJAX GET para o arquivo `read_alunos_turma.php`, utilizando a turma como parâmetro. Ou, para um servidor liberar um ou mais alunos, é feita uma requisição AJAX POST para o arquivo `insert_sepae_libera.php`, utilizando o ID do aluno, o motivo da liberação e o e-mail do servidor como parâmetros.

3.3. Web Service/API

O *Web Service* interage com o banco de dados por meio de uma função de conexão em um arquivo PHP que especifica servidor, usuário, senha e nome do banco de dados utilizado. Esse arquivo é requisitado em todos os arquivos do *Web Service*, que chamam a função para estabelecer uma conexão com o banco de dados para realizar consultas.

```

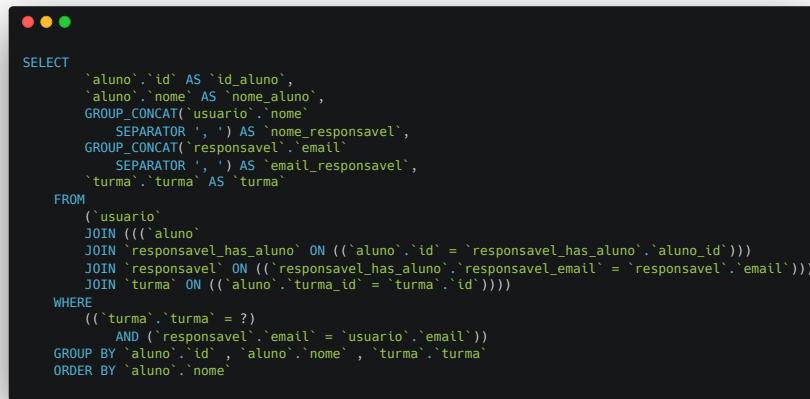
● ● ●

SELECT
    `recado`.`id` AS `id`,
    `recado`.`titulo` AS `titulo`,
    `recado`.`recado` AS `recado`,
    `recado`.`data` AS `data`,
    `recado`.`validade` AS `validade`,
    `recado`.`sepae_email` AS `sepae_email`
FROM
    `recado`
WHERE
    ((CURDATE() < `recado`.`validade`)
        OR (`recado`.`validade` IS NULL))
ORDER BY `recado`.`id` DESC

```

Figura 10. Consulta para leitura de recados

O arquivo `read_recados.php`, presente no *Web Service*, possui uma consulta responsável pela leitura dos recados válidos. Essa consulta, expressa pela Figura 10, possui condições que influenciam na exibição dos dados presentes no banco de dados. Por exemplo, o registro só sera exibido caso a data presente no campo `validade` seja posterior a data atual ou caso o campo `validade` for vazio.



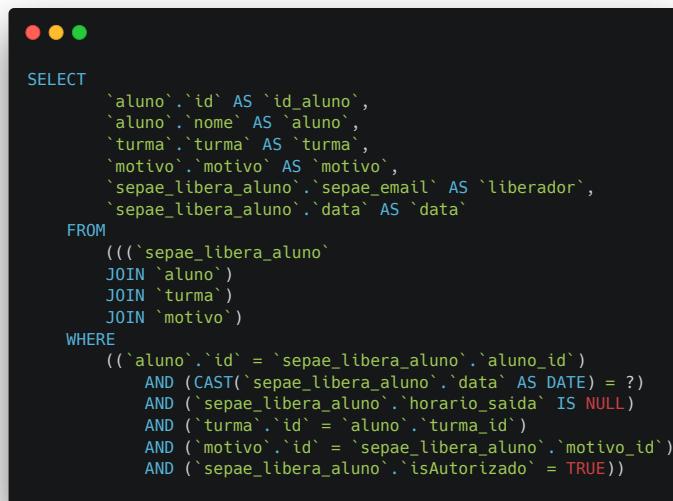
```

SELECT
    `aluno`.`id` AS `id_aluno`,
    `aluno`.`nome` AS `nome_aluno`,
    GROUP_CONCAT(`usuario`.`nome`
        SEPARATOR ', ') AS `nome_responsavel`,
    GROUP_CONCAT(`responsavel`.`email`
        SEPARATOR ', ') AS `email_responsavel`,
    `turma`.`turma` AS `turma`
FROM
    (`usuario`
    JOIN (((`aluno`
    JOIN `responsavel_has_aluno` ON ((`aluno`.`id` = `responsavel_has_aluno`.`aluno_id`)))
    JOIN `responsavel` ON ((`responsavel_has_aluno`.`responsavel_email` = `responsavel`.`email`)))
    JOIN `turma` ON ((`aluno`.`turma_id` = `turma`.`id`)))
WHERE
    ((`turma`.`turma` = ?)
        AND (`responsavel`.`email` = `usuario`.`email`))
GROUP BY `aluno`.`id`, `aluno`.`nome`, `turma`.`turma`
ORDER BY `aluno`.`nome`

```

Figura 11. Consulta para leitura de alunos por turma

A consulta encarregada pela leitura de alunos por turma no arquivo `read_alunos_turma.php`, expressa na Figura 11, é estruturada para retornar os dados de maneira ideal para o sistema. Por exemplo, o campo `nome_aluno` se repetiria caso o aluno tivesse mais de um responsável associado a ele e tais dados fossem exibidos em campos distintos. Diante disso, os respectivos dados `nome_responsavel` e `email_responsavel` de responsáveis associados a um mesmo aluno foram agrupados em um mesmo campo por meio da função `Group_Concat()`.



```

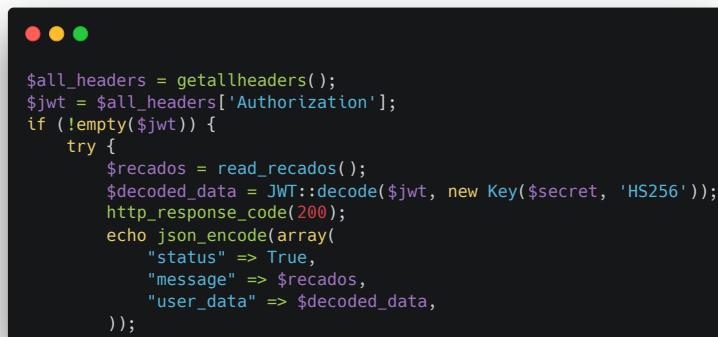
SELECT
    `aluno`.`id` AS `id_aluno`,
    `aluno`.`nome` AS `aluno`,
    `turma`.`turma` AS `turma`,
    `motivo`.`motivo` AS `motivo`,
    `sepae_libera_aluno`.`sepae_email` AS `liberador`,
    `sepae_libera_aluno`.`data` AS `data`
FROM
    (((`sepae_libera_aluno`
    JOIN `aluno`)
    JOIN `turma`)
    JOIN `motivo`)
WHERE
    ((`aluno`.`id` = `sepae_libera_aluno`.`aluno_id`)
        AND (CAST(`sepae_libera_aluno`.`data` AS DATE) = ?)
        AND (`sepae_libera_aluno`.`horario_saida` IS NULL)
        AND (`turma`.`id` = `aluno`.`turma_id`)
        AND (`motivo`.`id` = `sepae_libera_aluno`.`motivo_id`)
        AND (`sepae_libera_aluno`.`isAutorizado` = TRUE))

```

Figura 12. Consulta para leitura de alunos liberados

As consultas utilizadas pelo arquivo `read_alunos_liberados.php` permitem a leitura de liberações, pela SEPAE ou por responsáveis. A Figura 12 expõe a consulta para liberações pela SEPAE, que difere das por responsáveis apenas pelo nome das colunas e pela condição `isAutorizado`. Nessa consulta, os registros só são exibidos caso o campo `horario_saida` seja vazio, o campo `data` represente o dia da consulta e o campo `isAutorizado` for verdadeiro. Desse modo, somente as liberações diárias, autorizadas pelos responsáveis em que o aluno ainda não saiu do instituto são exibidas.

Além disso, para a *API*, as consultas do *Web Service* foram transformadas em funções PHP, que retornam um objeto do tipo JSON. A Figura 13 expõe um trecho da *API* para requisitar recados, que chama a função `read_recados()`.



```
$all_headers = getallheaders();
$jwt = $all_headers['Authorization'];
if (!empty($jwt)) {
    try {
        $recados = read_recados();
        $decoded_data = JWT::decode($jwt, new Key($secret, 'HS256'));
        http_response_code(200);
        echo json_encode(array(
            "status" => True,
            "message" => $recados,
            "user_data" => $decoded_data,
        ));
    }
}
```

Figura 13. Decodificação do Token e retorno

O *token* JWT é resgatado a partir do cabeçalho da requisição enviada pelo cliente e decodificado. Se o *token* for válido, o servidor retorna uma resposta HTTP 200 contendo os recados, caso o contrário, ele retorna uma resposta HTTP 500. Também foram criadas funções que recebem parâmetros utilizados nas consultas, como a função `insert_responsavel_libera()`, que recebe e-mail do responsável, identificação do aluno liberado e motivo da liberação.

A API também é utilizada para realizar a validação do e-mail do responsável. Tal processo ocorre da seguinte forma: o cliente — nesse caso, o responsável, por intermédio do aplicativo — manda uma requisição do tipo POST para o endpoint `send_verification_email.php`. Esse arquivo, por sua vez, envia um código de seis dígitos para o e-mail contido na requisição, através da biblioteca PHP Mailer e o serviço SMTP do Gmail. O código de verificação e a data e hora em que foi criado são inseridos no banco de dados e associados ao e-mail do responsável.

Após o envio bem-sucedido do e-mail com o código de verificação, o usuário é direcionado para uma tela em que deve digitar o código recebido. Ao clicar em enviar, uma requisição contendo o código digitado e o e-mail do responsável é feita para o endpoint `validate_verification_code.php`. Esse arquivo verifica se o código digitado existe no banco de dados, e se o horário em que a requisição foi feita não excede em 15 minutos o horário em que o código foi criado.

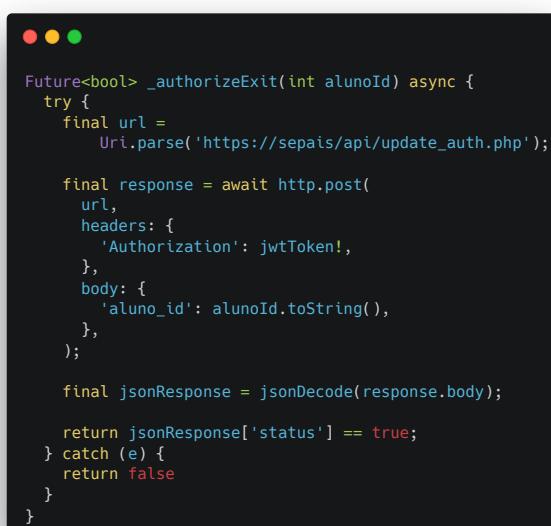
Caso o código enviado seja válido, o usuário é direcionado para uma tela em que deve digitar sua nova senha. Ao clicar em enviar, uma requisição contendo o código digitado, o e-mail do responsável e a senha é feita para o *endpoint* register_password.php. Esse arquivo cria um *hash* da senha e a atualiza no registro do responsável. Dessa forma, criam-se as credenciais de login que o responsável irá utilizar no aplicativo;

Além disso, a API também é utilizada para auxiliar no envio de notificações. Nesse sentido, para que seja possível implementar tal funcionalidade, é necessário que se tenha um *token* associado ao dispositivo móvel de cada responsável. Essa informação é armazenada no banco através do arquivo update_token.php. Toda vez em que se cria ou atualiza um *token* no aplicativo, uma requisição é feita para esse arquivo, contendo o token JWT no cabeçalho — como é de praxe para assegurar a integridade das requisições — e o *token* do aplicativo móvel no corpo da requisição. Ao receber a requisição, o arquivo atualiza o token no banco de dados.

3.4. Aplicativo SEPAIS

O aplicativo objetiva centralizar as ações realizadas pelos responsáveis nos processos de liberação, autorização e visualização de recados. Dessa forma, evita-se a necessidade de comunicação textual com a SEPAE por meio de aplicativos como o Whatsapp.

Conforme citado anteriormente, a autenticação dos responsáveis no aplicativo será feita através da tecnologia JSON Web Token (JWT). O usuário manda, através de uma requisição HTTP do tipo POST, suas credenciais para a API. Se o enviado coincidir com as informações armazenadas no banco de dados, o Servidor retorna para o Cliente um *token*, que será armazenado localmente por meio do *plugin shared_preferences*. Este *token* deverá ser enviado junto de todas as requisições subsequentes, a fim de garantir que o autor das requisições esteja autenticado, e também para recuperar dados do cliente. O *token* possui uma data de validade, que ao expirar, o usuário deverá fazer login novamente.



```
Future<bool> _authorizeExit(int alunoId) async {
  try {
    final url =
        Uri.parse('https://sepais/api/update_auth.php');

    final response = await http.post(
      url,
      headers: {
        'Authorization': jwtToken!,
      },
      body: {
        'aluno_id': alunoId.toString(),
      },
    );

    final jsonResponse = jsonDecode(response.body);

    return jsonResponse['status'] == true;
  } catch (e) {
    return false
  }
}
```

Figura 14. Função para autorizar saída de aluno

A utilização do JWT é de fundamental importância ao passo que além de autenticar o usuário, também permite que, dados dele sejam resgatados e utilizados nas consultas requisitadas. Na Figura 14 a função `_authorizeExit()` é mostrada, onde o *token* é enviado no cabeçalho da requisição — no qual o e-mail do responsável será recuperado —, e a identificação do aluno é enviada no corpo da requisição. Essa função é utilizada para autorizar a saída de um aluno que já foi liberado pela SEPAE.

Ademais, outras funções foram desenvolvidas para verificar ou alterar o estado de cada dependente, a saber:

- `_fetchDependents()`, para requisitar os dependentes do responsável logado;
- `_fetchMessages()`, para requisitar os recados enviados pela SEPAE;
- `_liberateDependent()` para um responsável liberar um dependente;
- `_fetchLiberatedDependents()`, para requisitar os dependentes do usuário logado que foram liberados pela SEPAE;
- `_fetchAuthorizedDependents()` para requisitar os dependentes que foram autorizados a sair pelos responsáveis;
- `_fetchExitTime()` para verificar o horário de saída de um dependente.

As últimas três funções supracitadas são chamadas periodicamente, a cada três segundos, a fim de garantir que o estado do aluno esteja sempre atualizado de acordo com o banco de dados, de forma dinâmica.

Para realizar a funcionalidade de envio de notificações utiliza-se o *Firebase Cloud Messaging*, um serviço de nuvem multiplataforma que permite que aplicativos de terceiros enviem notificações para usuários finais gratuitamente. Após registrar o aplicativo no serviço, obtém-se um *endpoint* de API exclusivo, que é utilizado para enviar as notificações. Cada responsável, ao entrar no aplicativo e autorizar o envio de notificações, gera um *token* de registro único, associado ao seu dispositivo, que é utilizado pelo FCM para direcionar as mensagens. Este *token* é armazenado no banco de dados a partir da API, conforme explicado anteriormente.

Nesse sentido, a partir das interações realizadas na aplicação web, requisições são feitas para esse *endpoint* a fim de que os responsáveis sejam notificados de determinadas alterações. Como por exemplo, ao liberar uma turma, o sistema percorre todos os responsáveis dos alunos da turma em questão, e a partir das informações registradas no banco de dados, como o nome do aluno e o token do responsável, faz requisições para o *endpoint* supracitado, que realiza o trabalho de notificar os responsáveis de que seus dependentes foram liberados.

Após login bem-sucedido, os responsáveis terão acesso a uma relação dos seus dependentes e suas respectivas turmas. A depender do estado destes alunos, diferentes ações poderão ser tomadas. Caso ele esteja em aula, o usuário poderá liberá-lo, selecionando o motivo da liberação. Por outro lado, caso a SEPAE tenha liberado o dependente, o responsável poderá autorizar a liberação. Por fim, quando o discente passar pela portaria e sair da instituição, o responsável poderá visualizar o horário que ele saiu. Além disso, também é possível visualizar os recados enviados pela Seção Pedagógica. A Figura 15 exibe as principais telas do aplicativo.

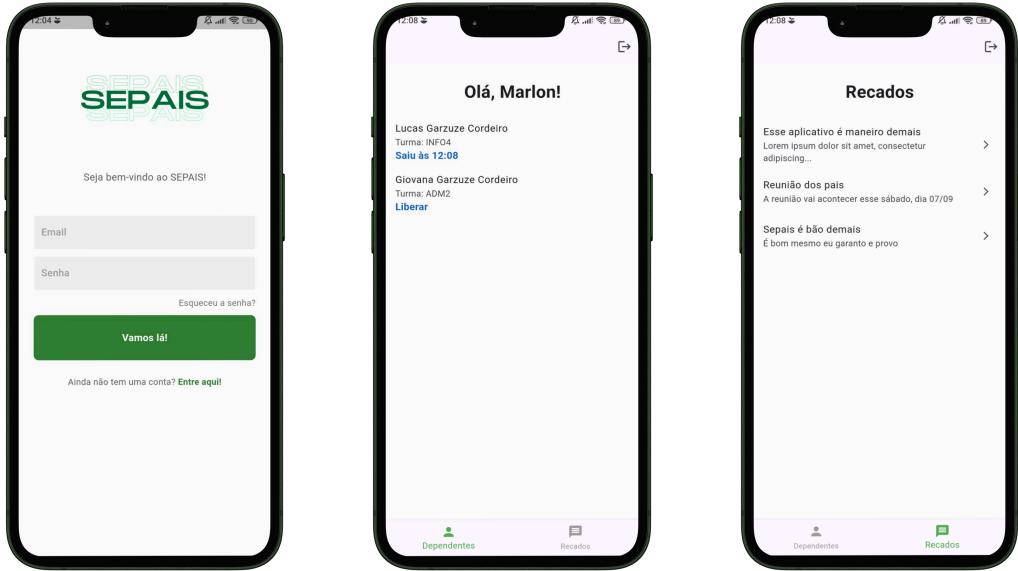


Figura 15. Principais telas do Aplicativo SEPAIS.

4. Conclusão

Ao analisar os resultados obtidos na Seção 3, é possível concluir que o SEPAIS apresenta os seguintes objetivos cumpridos: um banco de dados modelado em SQL para o armazenamento de informações; uma interface web que auxilia os funcionários na liberação dos alunos; funções do *Web Service* e da *API* que permitem que o responsável acompanhe o processo de liberação do seu dependente; uma interface desenvolvida em Flutter que permite que os responsáveis realizem liberações justificadas de seus dependentes.

Portanto, o SEPAIS cumpriu todos os objetivos específicos estabelecidos inicialmente e implementou funcionalidades adicionais como o registro de atrasos, o sistema de recados e a visualização de históricos.

As contribuições esperadas com a implementação do SEPAIS têm o potencial de transformar a dinâmica operacional e a comunicação dentro do Instituto Federal do Paraná. Primeiramente, espera-se uma melhoria substancial na comunicação institucional, facilitando a interação entre a Seção Pedagógica e de Assuntos Estudantis, os responsáveis pelos alunos e os servidores da portaria.

Ademais, como desenvolvimento futuro com o propósito de aumentar a acessibilidade e facilitar a utilização do aplicativo — principalmente no que tange usuários idosos —, as funcionalidades do aplicativo podem ser descritas por meio de um tutorial interativo, que explica como utilizar o sistema. Além disso, também é possível implementar a produção de conteúdo offline, como materiais impressos, que auxiliem os responsáveis pouco familiarizados com a tecnologia a utilizar o aplicativo.

Referências

- [AgendaEdu 2024] AgendaEdu (2024). Software edugo. plataforma educacional desenvolvida pela agenda edu. <https://www.agendaedu.com/>. Acesso em 13 de set. de 2024.
- [Axmark et al. 2024] Axmark, D., Larsson, A., and Widenius, F. M. M. (2024). Mysql 8.0.33. Acesso em 13 de set. de 2024.
- [BioMessenger 2014] BioMessenger (2014). Biomessenger. Site e aplicativo móvel. Disponível em plataforma web e dispositivos móveis.
- [BRASIL 2008] BRASIL (2008). Lei nº 11.892, de 29 de dezembro de 2008. Diário Oficial da União. Aprovada em 29 de dezembro de 2008.
- [Genç 2017] Genç, R. (2017). The importance of communication in sustainability & sustainable strategies. *Procedia Manufacturing*, 8:511–516.
- [IBM 2024] IBM (2024). What is three-tier architecture? *IBM*.
- [IFPR 2021] IFPR (2021). *Guia do Estudante*. Instituto Federal do Paraná, Colombo. Acesso em 15 de mar. de 2024.
- [Jones et al. 2015] Jones, M., Bradley, J., and Sakimura, N. (2015). Rfc 7519: Json web token (jwt).
- [Juviler 2022] Juviler, J. (2022). What is a modal and when should i use one? <https://blog.hubspot.com/website/modal-web-design>. Acesso em 13 de set. de 2024.
- [Radovic Markovic e Salamzadeh 2018] Radovic Markovic, M. and Salamzadeh, A. (2018). The importance of communication in business management. In *Radovic Markovic, M., & Salamzadeh, A.(2018). The Importance of Communication in Business Management, The 7th International Scientific Conference on Employment, Education and Entrepreneurship, Belgrade, Serbia.*
- [Soares et al. 2022] Soares, T. P., da Silva, L. d. Q., Jung, H. S., and Fossatti, P. (2022). Gestão escolar e clima organizacional: a influência da comunicação. *Revista Educar Mais*, 6:372–381.
- [SpryMedia 2024] SpryMedia (2024). *DataTables: Advanced interaction controls for HTML tables*. Acesso em: 7 de out. de 2024.
- [Wathan et al. 2024] Wathan, A., Reinink, J., Hemphill, D., and Schoger, S. (2024). Tailwind css v3.3.2. <https://tailwindcss.com/>. Acesso em 13 de set. de 2024.