

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE BOTUCATU  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO  
DE SISTEMAS**

**MARCOS GASPARINI JUNIOR**

**AUTO SIGA: SISTEMA DE COLETA DE DADOS DO SIGA DE MANEIRA  
AUTOMATIZADA, INTEGRADO A SISTEMAS DE NOTIFICAÇÃO POR E-  
MAIL E POR WHATSAPP**

Botucatu – SP  
Novembro - 2024

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE BOTUCATU  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO  
DE SISTEMAS**

**MARCOS GASPARINI JUNIOR**

**AUTO SIGA: SISTEMA DE COLETA DE DADOS DO SIGA DE MANEIRA  
AUTOMATIZADA, INTEGRADO A SISTEMAS DE NOTIFICAÇÃO POR E-  
MAIL E POR WHATSAPP**

Orientador: Prof. Renato Luiz Gambarato

Projeto de Conclusão de Curso apresentado à  
FATEC - Faculdade de Tecnologia de  
Botucatu, para obtenção do título de  
Tecnólogo no Curso Superior de Análise e  
Desenvolvimento de Sistemas.

Botucatu - SP  
Novembro - 2024

## RESUMO

O acompanhamento das informações acadêmicas por estudantes da Fatec, como notas e frequência, ainda depende de consultas manuais constantes ao sistema SIGA, o que frequentemente causa lentidão e queda de desempenho, especialmente em períodos de alta demanda. Para resolver esse problema, foi desenvolvido o AutoSIGA, uma solução automatizada que utiliza Selenium para monitorar o SIGA e envia notificações proativas aos estudantes via WhatsApp e e-mail sempre que há atualizações em seus dados acadêmicos. A aplicação conta com uma interface desktop em Windows Forms, desenvolvida com C#, e uma API (Interface de Programação de Aplicações) em Node.js e Express para integração com o WhatsApp, utilizando a biblioteca WhatsappWebJS para envio das notificações. O armazenamento de dados é feito através do Entity Framework, com um banco de dados MySQL, proporcionando uma experiência de consulta mais eficiente e conveniente. Com isso, o AutoSIGA contribui para reduzir a necessidade de consultas manuais e alivia a carga no sistema SIGA.

**Palavras-chave:** Automação. Notificações acadêmicas. Selenium. Windows Forms. Express. WhatsappWebJS. Entity Framework. MySQL.

## SUMÁRIO

	Página
<b>1 INTRODUÇÃO .....</b>	<b>4</b>
<b>2. CARACTERIZAÇÃO DA TECNOLOGIA UTILIZADA .....</b>	<b>6</b>
2.1 Selenium .....	7
2.2 Express.....	7
2.3 Whatsapp Web JS .....	7
2.4 Postman .....	8
2.5 MySQL .....	8
2.6 Entity Framework .....	9
2.7 Windows Forms .....	9
2.8 Engenharia de Software.....	10
2.8 SIGA .....	11
<b>3 FERRAMENTAS DE DESENVOLVIMENTO .....</b>	<b>12</b>
<b>4 DESCRIÇÃO DO SOFTWARE .....</b>	<b>13</b>
4.1 Tela de Primeiro Acesso.....	13
4.2 Tela de Login.....	14
4.3 Tela Principal.....	15
4.4 Tela de Perfil .....	18
4.5 Tela de Matérias, Notas e Faltas .....	19
4.6 Tela de Preferências .....	21
4.41 Diagrama Banco de Dados.....	24
<b>5 APLICABILIDADE .....</b>	<b>25</b>
<b>REFERÊNCIAS .....</b>	<b>26</b>

## 1 INTRODUÇÃO

Atualmente, os estudantes da Fatec vivenciam uma rotina acadêmica em que o acompanhamento de informações, como notas e frequência, poderia ser significativamente aprimorado. Embora o sistema SIGA ofereça acesso a esses dados, muitos alunos recorrem a consultas manuais constantes apenas para verificar se há novas atualizações, frequentemente reclamando da baixa interatividade. Esse hábito, além de consumir tempo, contribui para picos de acesso ao sistema, podendo resultar em quedas de desempenho e lentidão no SIGA, especialmente durante períodos de alta demanda.

Para resolver esse problema, o projeto AutoSIGA foi desenvolvido como uma solução que automatiza a consulta ao SIGA e notifica os estudantes sempre que houver alterações relevantes em seus dados acadêmicos. A aplicação funciona de maneira totalmente local, oferecendo uma interface desktop para visualização das informações e utilizando o Selenium para monitorar o SIGA e enviar notificações proativas via WhatsApp e E-mail. Com isso, o AutoSIGA reduz a necessidade de consultas manuais frequentes, aliviando a carga no sistema oficial da instituição.

Este trabalho utiliza tecnologias bem estabelecidas no mercado de desenvolvimento de software e a automação de consultas é possibilitada principalmente pelo Selenium, uma ferramenta originalmente usada para testes, mas que aqui foi adaptada para monitoramento e extração de dados. O desenvolvimento da interface desktop foi realizado com Windows Forms em C#, e, para a mensageria via WhatsApp, foi desenvolvida uma API em Node.js e Express, responsável por enviar notificações ao WhatsApp do usuário. A persistência de dados para consulta offline é feita com MySQL, utilizando o Entity Framework, garantindo

que os dados sejam armazenados de maneira eficiente e permitindo o acesso rápido às informações mesmo quando o sistema não estiver conectado à internet.

Assim, o objetivo do AutoSIGA é não apenas notificar os estudantes sobre atualizações importantes relacionadas à sua vida acadêmica, mas também oferecer um código bem estruturado e de fácil entendimento, que sirva de referência para estudantes interessados na área. O sistema foi inteiramente desenvolvido com boas práticas de programação amplamente utilizadas no mercado de trabalho, permitindo que esses estudantes possam não apenas utilizar o AutoSIGA, mas também contribuir para sua evolução. Dessa forma, o AutoSIGA visa não só aprimorar o acesso a dados acadêmicos, mas também ser uma plataforma que estimule o aprendizado e a aplicação prática de técnicas de automação em diferentes setores acadêmicos.

## 2. CARACTERIZAÇÃO DA TECNOLOGIA UTILIZADA

No ambiente acadêmico, é comum observar a ansiedade dos estudantes ao aguardar atualizações de notas e frequência, que nem sempre são informadas de forma proativa. Para resolver esse problema, o AutoSIGA foi desenvolvido para notificar os usuários sempre que houver atualizações relevantes. A solução utilizou diversas tecnologias para criar uma ferramenta eficiente e fácil de usar.

O projeto foi desenvolvido utilizando uma combinação de tecnologias para proporcionar uma solução eficiente e acessível, o Selenium foi escolhido para automatizar o monitoramento do SIGA, realizando consultas periódicas e detectando alterações de forma autônoma, a interface do AutoSIGA foi construída com Windows Forms, proporcionando um ambiente amigável para a visualização das informações acadêmicas extraídas do SIGA, a biblioteca WhatsAppWebJS foi integrada para permitir o envio de notificações instantâneas via WhatsApp, o *Back-end* do envio de mensagens foi desenvolvido com Express, enquanto o banco de dados MySQL armazena as informações localmente na aplicação Desktop, sem envio de dados externos. Para otimizar a interação com o banco de dados, o Entity Framework foi utilizado, garantindo flexibilidade e consistência e por fim o Postman foi empregado para testar a API do WhatsApp, garantindo que a integração fosse realizada corretamente e de forma eficiente entre a aplicação Desktop e a API de mensagem.

Essas tecnologias, aplicadas de maneira integrada, garantem que o AutoSIGA não só resolva a falta de notificações sobre atualizações acadêmicas, mas também ofereça uma solução prática e intuitiva, melhorando significativamente a experiência acadêmica dos estudantes.

## 2.1 Selenium

Selenium é um projeto que abrange uma variedade de ferramentas e bibliotecas que permitem e suportam a automação de navegadores da web (SELENIUM, 2024). A escolha do Selenium para o desenvolvimento do AutoSIGA foi fundamental para automatizar a coleta de dados acadêmicos de maneira eficiente e transparente. Através do Selenium, foi possível realizar o login no sistema SIGA de forma automatizada, sem que o usuário precisasse interagir diretamente com o navegador. Essa abordagem, que funciona de maneira semelhante a uma API, permite que o AutoSIGA acesse e capture os dados diretamente do SIGA em segundo plano, sem que o usuário perceba que o navegador está sendo acessado e realizando o login automaticamente. Dessa forma, o AutoSIGA oferece uma experiência fluida, onde o processo de captura e gerenciamento das informações acadêmicas ocorre de forma invisível para o usuário, otimizando a interação com o sistema e tornando-a mais eficiente.

## 2.2 Express

O Express é um framework minimalista e flexível para desenvolvimento de aplicativos web em Node.js, oferecendo um conjunto robusto de recursos para criação de APIs e aplicações voltadas para web e dispositivos móveis (EXPRESS, 2024). A escolha pelo Express para o desenvolvimento do AutoSIGA foi motivada pela necessidade de uma solução ágil para o envio de notificações via WhatsApp. Em conjunto com a biblioteca WhatsApp Web JS, o Express permitiu a construção de uma API que possibilita a comunicação entre diferentes componentes de software por meio de um conjunto de definições e protocolos estabelecidos (AMAZON, 2024). Essa API facilita a comunicação entre o AutoSIGA e o WhatsApp, de modo que, após a detecção de uma atualização no SIGA, o sistema direciona automaticamente uma mensagem para o WhatsApp do usuário, garantindo notificações em tempo real e reduzindo a necessidade de verificações manuais.

## 2.3 Whatsapp Web JS

A biblioteca WhatsApp Web JS é uma ferramenta de automação projetada para interagir com o WhatsApp Web, o cliente da API do WhatsApp se conecta por meio do aplicativo do navegador WhatsApp Web, acessando suas funções internas. Isso lhe concede



acesso a quase todos os recursos disponíveis no WhatsApp Web (WHATSAPPWEBJS, 2024). A decisão de utilizá-la no AutoSIGA apesar dos riscos, a decisão foi baseada totalmente em sua capacidade de fornecer um serviço de notificação via WhatsApp totalmente gratuito, possibilitando que os usuários recebam alertas sobre alterações acadêmicas diretamente em seus dispositivos móveis. Ao invés de depender de um número de telefone específico ou de uma conta de desenvolvedor no WhatsApp Business API, o WhatsApp Web JS permite que o AutoSIGA se conecte ao WhatsApp Web do próprio usuário, usando um código QR para autenticação e garantindo que o sistema envie notificações diretamente para seus contatos autorizados, existe também a possibilidade de que WhatsApp Web JS seja configurado com um número específico do robô de notificações, centralizando as mensagens automáticas e evitando interferências com o uso pessoal do aplicativo.

Essa abordagem, além de facilitar a implementação, elimina a necessidade de recursos adicionais de infraestrutura, tornando o processo de notificação via WhatsApp mais acessível e ágil para estudantes que utilizam o AutoSIGA.

## **2.4 Postman**

O Postman foi uma ferramenta essencial no desenvolvimento do AutoSIGA, especialmente para garantir a funcionalidade da API construída com Express em conjunto com o WhatsApp Web JS. A necessidade do Postman surgiu ao longo do processo de criação e validação da API de notificação, possibilitando testes detalhados das rotas e endpoints responsáveis pelo envio de notificações via WhatsApp. Com o Postman, foi possível simular requisições HTTP, verificar o comportamento do servidor e assegurar que as mensagens fossem enviadas corretamente para os números de WhatsApp autorizados. Além disso, a ferramenta permitiu monitorar e corrigir potenciais erros de comunicação entre os componentes da aplicação, garantindo uma integração eficiente e confiável entre o Express e o WhatsApp Web JS. A capacidade de realizar testes contínuos com o Postman contribuiu significativamente para a estabilidade e qualidade do sistema, assegurando que os processos de notificação fossem executados conforme esperado.

## **2.5 MySQL**

A escolha do MySQL para o AutoSIGA se fundamenta principalmente na

necessidade de armazenar os dados para poder realizar consultar mesmo que offline, e secundariamente em sua robustez, flexibilidade e ampla adoção no mercado como um sistema de gerenciamento de banco de dados relacional de alto desempenho. O MySQL oferece uma estrutura sólida para armazenar de maneira confiável as informações necessárias ao funcionamento do sistema, como dados de usuários, configurações e preferências personalizadas para notificações, garantindo consistência e integridade dos dados.

Além disso, o MySQL proporciona escalabilidade e eficiência na execução de consultas, o que é essencial para o AutoSIGA processar e armazenar informações rapidamente, sem comprometer o desempenho da aplicação. Com uma arquitetura que facilita a manutenção e expansão do sistema, o MySQL permite que o AutoSIGA responda a futuras necessidades dos usuários, integrando-se eficientemente com as outras tecnologias utilizadas no projeto e permitindo uma experiência de uso estável e confiável.

## **2.6 Entity Framework**

A escolha do Entity Framework para o desenvolvimento do AutoSIGA foi motivada pela necessidade de um banco de dados local, que armazenasse as configurações e preferências do usuário de maneira prática e acessível, ele basicamente é um mapeador moderno de relação de objetos que permite criar uma camada de acesso a dados limpa, portátil e de alto nível com o .NET (C#) em vários bancos de dados (MICROSOFT, 2024). Como a aplicação foi desenvolvida em Windows Forms, o Entity Framework mostrou-se uma solução ideal, integrando-se de forma eficiente ao ambiente .NET e permitindo que os dados fossem gerenciados localmente, sem a necessidade de servidores externos. Essa abordagem oferece ao usuário maior controle e segurança sobre suas informações, visto que tudo fica armazenado em seu próprio dispositivo. Além disso, a facilidade de configuração e a compatibilidade com C# permitiram uma integração rápida e uma curva de aprendizado reduzida, essencial para otimizar o processo de desenvolvimento. Dessa forma, o Entity Framework contribuiu para uma solução robusta e ágil, alinhada às necessidades de armazenamento local do AutoSIGA.

## **2.7 Windows Forms**

A escolha do Windows Forms para o desenvolvimento do AutoSIGA foi orientada pela robustez que ele oferece ao desenvolvimento de aplicações desktop no ecossistema .NET

e por seu alinhamento com o nicho de soluções locais e dedicadas a Desktop. Sendo uma tecnologia amplamente testada e consolidada, o Windows Forms é descrito como "uma estrutura de interface do usuário para a criação de aplicativos da área de trabalho do Windows. Ele fornece uma das maneiras mais produtivas de criar aplicativos da área de trabalho com base no designer visual fornecido no Visual Studio. A funcionalidade, como o posicionamento de arrastar e soltar de controles visuais, facilita a criação de aplicativos da área de trabalho" (MICROSOFT, 2024). Essas características são ideais para aplicações de automação de dados que exigem estabilidade e desempenho.

Além disso, o Windows Forms oferece um conjunto abrangente de ferramentas e controles que facilitam a criação de interfaces interativas e responsivas, essenciais para garantir uma experiência satisfatória ao usuário. Sua compatibilidade direta com o Entity Framework e outras bibliotecas .NET permite que o AutoSIGA gerencie dados e interaja com o sistema de forma otimizada, oferecendo uma solução desktop confiável e adequada ao contexto acadêmico e operacional.

## **2.8 Engenharia de Software**

A gestão de atividades no desenvolvimento do AutoSIGA foi orientada pelo uso de uma estrutura que organizasse e controlasse de forma eficiente o fluxo de tarefas ao longo do ciclo de vida do projeto. Através do conceito de *to-do lists*, foi possível garantir que cada tarefa fosse devidamente registrada, acompanhada e concluída dentro do cronograma estabelecido. A abordagem adotada facilitou a visualização das atividades e ajudou a priorizar os processos mais urgentes, tornando o desenvolvimento mais controlado e organizado.

Além disso, a diagramação das classes foi realizada na ferramenta Astah, a qual permitiu a criação de diagramas claros e precisos que representam as relações entre as entidades e suas interações no sistema. O uso dessa ferramenta foi essencial para estruturar o código de maneira lógica, definindo responsabilidades e visualizando as interações entre objetos e dados de forma intuitiva.

Com a diagramação adequada, foi possível mapear as dependências e fluxos do sistema, garantindo uma arquitetura mais coesa e facilmente manutenível. A utilização de ferramentas específicas de gestão de tarefas, como listas de tarefas, junto com a diagramação das classes no Astah, forneceu uma estrutura lógica e visual que não apenas contribuiu para o controle das atividades, mas também ajudou a documentar a evolução do sistema de

maneira clara e acessível. Esse controle de atividades, aliado à diagramação das classes, foi vital para assegurar que o projeto fosse entregue de maneira eficiente, com código bem estruturado e de acordo com os requisitos planejados.

## **2.8 SIGA**

O AutoSIGA foi desenvolvido com o propósito de automatizar funções essenciais que o Sistema Integrado de Gestão Acadêmica (SIGA, 2024), a plataforma acadêmica oficial, não cobre integralmente, proporcionando uma experiência mais dinâmica e prática para estudantes. Por meio de um sistema de coleta de dados que opera diretamente no ambiente do SIGA, o AutoSIGA é capaz de monitorar atualizações de notas, prazos de integralização, e outros dados relevantes, realizando a extração de informações em tempo real sem a necessidade de intervenções manuais constantes.

Essa integração permite que o AutoSIGA atue como uma extensão das funcionalidades do SIGA, garantindo que os estudantes sejam notificados automaticamente sobre novas publicações ou mudanças em suas informações acadêmicas. Em vez de os usuários precisarem acessar o SIGA frequentemente para verificar se há atualizações, o AutoSIGA se encarrega dessa tarefa e utiliza o WhatsApp Web JS para enviar notificações diretamente para o celular do usuário, facilitando o acesso rápido às informações.

Ao automatizar a coleta de dados diretamente no SIGA, o AutoSIGA visa preencher as lacunas deixadas pela plataforma oficial, oferecendo aos estudantes, especialmente aos líderes de sala, uma ferramenta que melhora significativamente o acompanhamento acadêmico e a comunicação de atualizações em tempo real.

### 3 FERRAMENTAS DE DESENVOLVIMENTO

O presente trabalho foi utilizado para desenvolvimento do software os seguintes materiais:

Computador com as seguintes configurações:

- Sistema Operacional: Windows (*64 bits*)
- Processador: *I5 7400*
- Memória: 8 GB de *RAM*
- Placa de vídeo: *GTX 1050ti*
- Armazenamento: 1 TB de espaço disponível.

Como ferramentas para o desenvolvimento do aplicativo foi utilizado os seguintes softwares:

- Node.js: para executar o ambiente de servidor Javascript.
- Visual Studio: IDE utilizada para desenvolvimento Windows forms e Selenium.
- Dbeaver: para servidor de interface com o Banco de dados MySQL.
- Git: para controle de versão do código-fonte.
- Visual Studio Code: como IDE para desenvolvimento API em Express.
- Postman: ferramenta para testar API do Whatsapp com Express

## 4 DESCRIÇÃO DO SOFTWARE

### 4.1 Tela de Primeiro Acesso

Figura 1 – Tela de Primeiro Acesso

Messageiro do SIGA

### Login Siga

CPF

Senha

Entrar

[Primeiro Acesso](#)

### Preferências

Whatsapp

E-mail

Atualizações por Whatsapp

Atualizações por E-mail

Figura 2 – Aviso de dados incorretos

Messageiro do SIGA

### Atenção

⚠️ Usuário ou senha incorretos. Por favor, verifique se suas credenciais são as mesmas do SIGA e tente novamente.

Ok

A tela de primeiro acesso, ilustrada na Figura 1, oferece campos de preenchimento obrigatório para o primeiro acesso:

- CPF: Campo para inserir o CPF do usuário cadastrado no SIGA.
- Senha: Campo para inserir a senha do usuário cadastrado no SIGA.
- Botão Entrar: Ao clicar neste botão, o sistema vai até o siga e valida se o usuário e senha correspondem, e se corresponderem os dados do usuário são previamente coletados, e é liberado o acesso completo ao sistema de coleta de dados, caso contrário uma notificação de senha incorreta aparece na tela ( Figura 2).
- Esqueceu a senha?: Link para a página de redefinição de senha, permitindo ao usuário recuperar o acesso caso tenha esquecido a senha.
- Criar Conta: Link para a página de registro, onde novos usuários podem criar uma conta.

## 4.2 Tela de Login

Figura 3 – Tela de login

Mensageiro do SIGA

### Login Siga

CPF

Senha

Entrar

[Primeiro Acesso](#)

**Fatec**  
Faculdade de Tecnologia

A tela de login, ilustrada na Figura 1, oferece a possibilidade de logar com os dados que já foram salvos previamente pelo aplicativo na tela de primeiro acesso (Figura 1):

- CPF: Campo para inserir o CPF quando o usuário já tiver acessado uma primeira vez.
- Senha: Campo para inserir a Senha quando o usuário já tiver acessado uma primeira vez.

### 4.3 Tela Principal

Figura 4 – Tela Principal

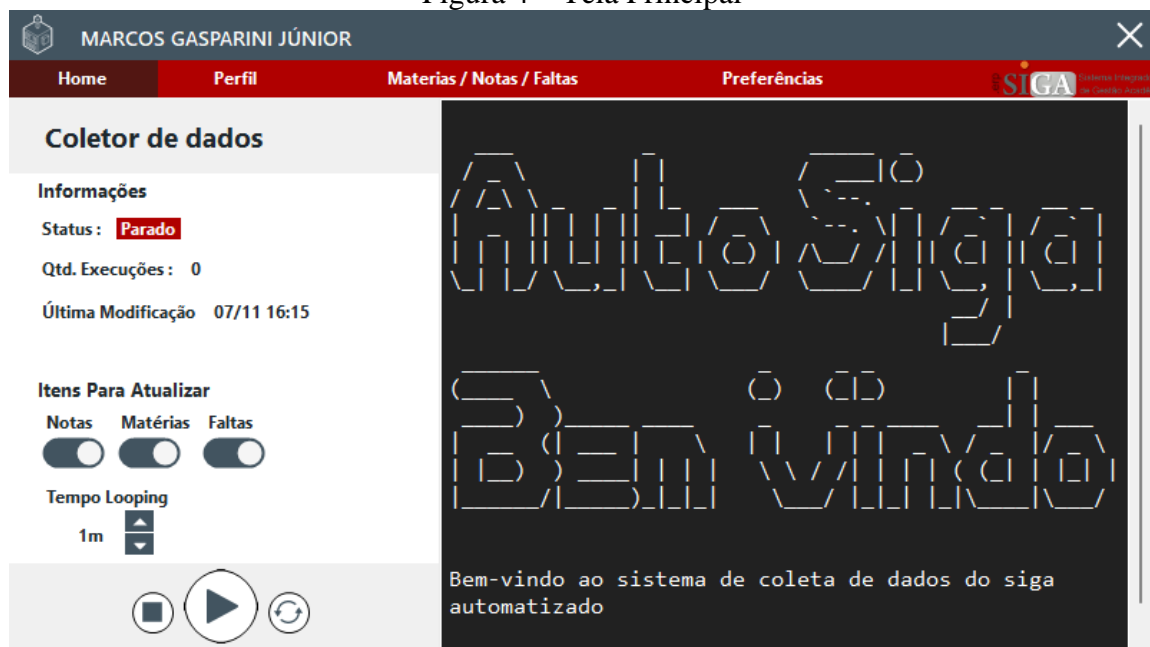




Figura 5 – Tela principal em execução

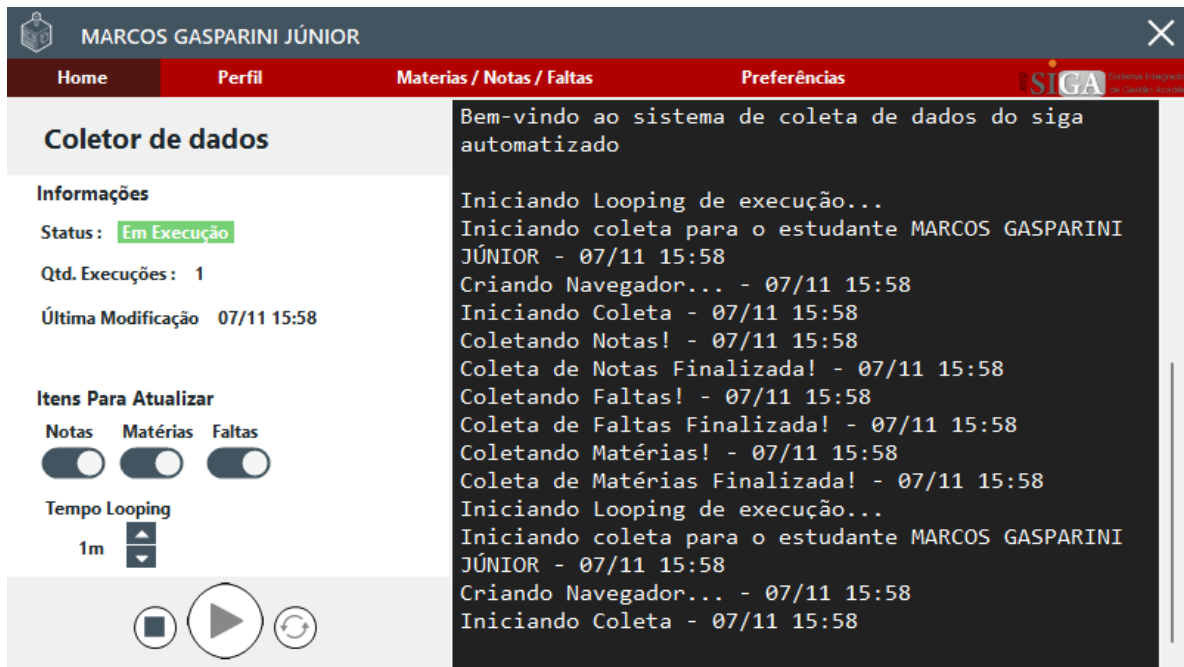


Figura 6 – Aviso de robô em Execução



Figura 7 – Minimização para a bandeja do sistema



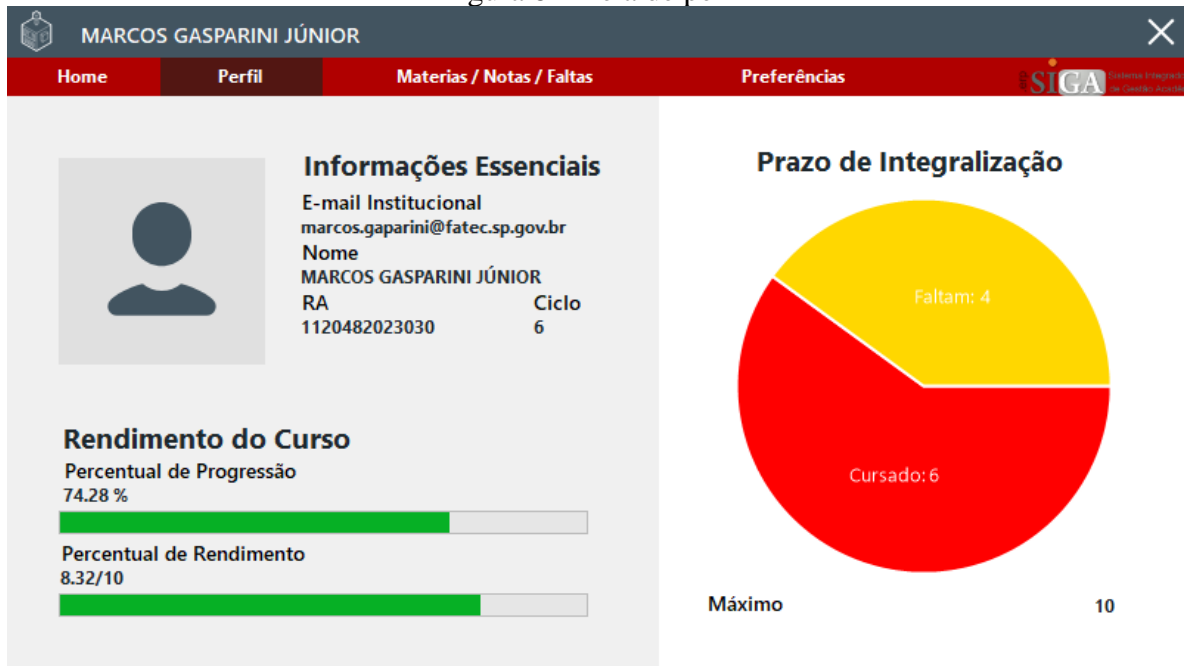
A tela principal, ilustrada na Figura 4, oferece as principais funcionalidades do sistema que consiste basicamente na coleta de dados do SIGA:

- Rótulo Status: Informa qual o status de execução do robô, que pode ser “Parado” ou “Em Execução”.
- Rótulo Última atualização: Informa a data da última atualização capturada pelo Robô.
- Rótulo Quantidade de Execuções: Informa a quantidade de execuções que o Robô executou, funciona como um contador.
- Caixa de Seleção Notas: Habilita o robô para coletar as notas do aluno.
- Caixa de Seleção Matérias: Habilita o robô para coletar as matérias do aluno.
- Caixa de Seleção Faltas: Habilita o robô para coletar as faltas do aluno.
- Seleção de Tempo Looping: Configura o tempo de espera até a próxima execução para quando o robô é posto para capturar dados em looping.
- Botão Executar: Executa a coleta de dados apenas uma vez.
- Botão Parar: Para a coleta de dados
- Botão Executar em Looping: Executa a coleta de dados infinitas vezes.

Quando a aplicação entra em execução, o terminal a direita será o responsável por apresentar para o usuário todos os *logs* gerados durante a execução do robô (Figura 5), e em caso de o usuário tentar fechar a aplicação enquanto o robô realiza a coleta de dados, uma janela de aviso será exibida, informando que o processo está em execução (Figura 6). Nesta janela, o usuário terá a opção de minimizar o aplicativo para a bandeja do sistema, permitindo que a janela seja ocultada enquanto o robô finaliza a coleta, ou cancelar a execução e fechar a aplicação. Para restaurar a janela para a área de trabalho, basta ir até o ícone do robô na bandeja do sistema, clicar com o mouse e selecionar a opção de voltar (Figura 7).

#### 4.4 Tela de Perfil

Figura 8 – Tela de perfil



A tela de perfil, ilustrada na Figura 8, oferece funcionalidades informativas e ilustrativas sobre a situação atual do estudante logado, apresentando as informações do SIGA de maneira mais dinâmica e autoexplicativa.

- Rótulo de percentual de progressão: Percentual de Progressão que equivale ao percentual realizado no curso pela carga horária das disciplinas aprovadas.
- Rótulo de percentual de rendimento: Percentual de Rendimento que equivale a nota média do estudante no curso.
- Área Informações Essenciais: Área responsável por apresentar os dados básicos do estudante, como E-mail institucional, Nome, RA, e o Ciclo que o estudante se encontra.
- Gráfico de Prazo de Integralização: Um gráfico que exhibe o progresso atual do estudante no curso, indicando o que já foi concluído e o que ainda falta para a conclusão. A soma dessas duas partes corresponde ao valor total necessário para a integralização do curso.

## 4.5 Tela de Matérias, Notas e Faltas

Figura 9 – Matérias, Notas e Faltas

MARCOS GASPARINI JÚNIOR

Home Perfil **Materias / Notas / Faltas** Preferências SIGA Sistema Integrado de Gestão Acadêmica

**Matérias / Notas / Faltas** Última Atualização - 07/11/2024 - 15:58 **Ciclo 6**

Matéria

🔍 🧹 Limpar Dados

NOME	PROFESSOR	P1	P2	P3	AUSÊNCIAS	MAX. AUSÊNCIAS
Gestão de Projetos	RICARDO RALL	0	0	0	8	20
Gestão de Equipes	LETÍCIA NEGRIS...	0	0	0	6	10
Empreendedorismo	JOSÉ RAFAEL P...	7.5	0	0	6	10
Ética e Responsabilidade Profissional	JOSÉ BENEDITO...	0	0	0	4	10
Auditoria de Sistemas (Escolha 3)	CIRO MARCOS ...	7.2	0	0	11	20
Tópicos Especiais em Informática (Escolha 2)	VIVIAN TOLEDO...	9.6	0	0	6	20
Gestão e Governança de Tecnologia da Informação	DIEGO AUGUST...	8.5	0	0	16	20
Inglês VI	MARCO ANTO...	0	0	0	6	10
Estágio Supervisionado	CARLOS ROBER...	0	0	0	0	20
Trabalho de Graduação II	ROGER CRISTHI...	0	0	0	0	20

Figura 10 – Matérias, Notas e Faltas com Filtro de busca preenchido

MARCOS GASPARINI JÚNIOR

Home Perfil **Materias / Notas / Faltas** Preferências SIGA Sistema Integrado de Gestão Acadêmica

**Matérias / Notas / Faltas** Última Atualização - 07/11/2024 - 16:11 **Ciclo 6**

Matéria

🔍 🧹 Limpar Dados

NOME	PROFESSOR	P1	P2	P3	AUSÊNCIAS	MAX. AUSÊNCIAS
Gestão de Projetos	RICARDO RALL	0	0	0	8	20
Gestão de Equipes	LETÍCIA NEGRIS...	0	0	0	6	10
Gestão e Governança de Tecnologia da Informação	DIEGO AUGUST...	8.5	0	0	16	20

Figura 11 – Matérias, Notas e Faltas não preenchidas



A tela matérias, notas e faltas conforme ilustrado na Figura 9, apresenta todos os dados das Matérias encontradas no SIGA de maneira resumida e simples em forma de tabela:

- Campo Matéria: Campo para inserir a Matéria pela qual o usuário quer buscar no sistema.
- Botão Buscar: Botão responsável por executar a busca preenchida no campo matéria (Figura 10).
- Botão Limpar Busca: Botão responsável por limpar a busca preenchida no campo matéria, trazendo todas as matérias de volta (Figura 9).
- Botão Limpar Dados: Botão responsável por limpar todos os dados de Matérias, Notas e Faltas do banco, para que caso o ciclo se encerre, a aplicação não armazene o histórico (Figura 11)

## 4.6 Tela de Preferências

Figura 12 – Tela de Primeiro Acesso

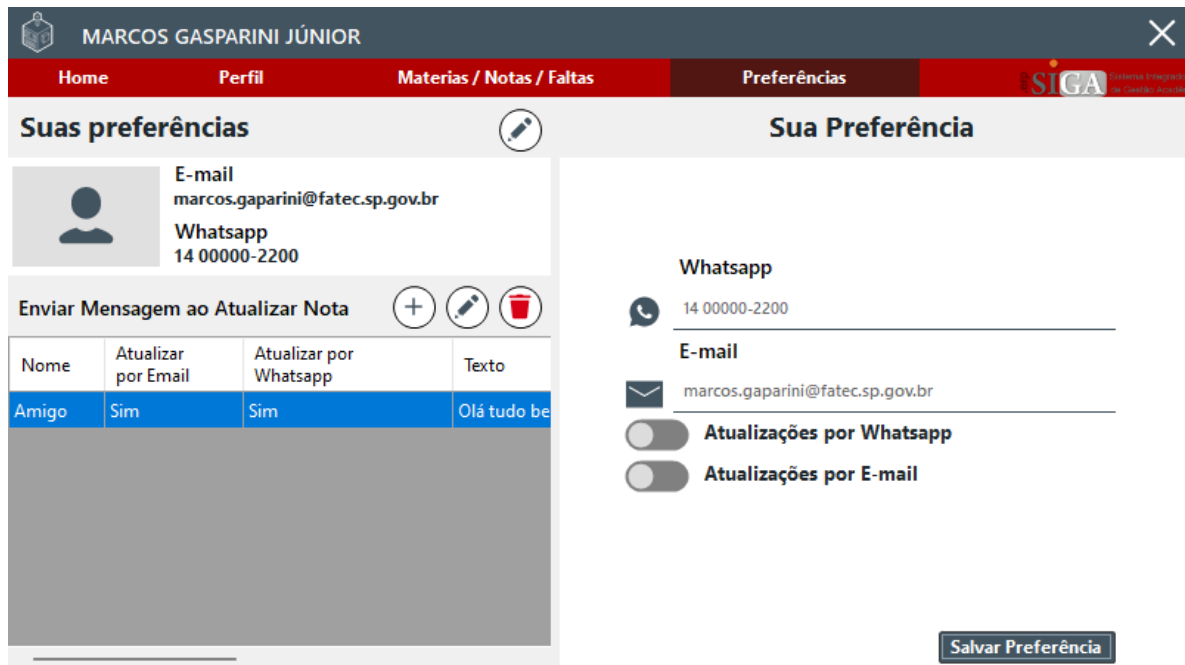


Figura 13 – Tela de Primeiro Acesso

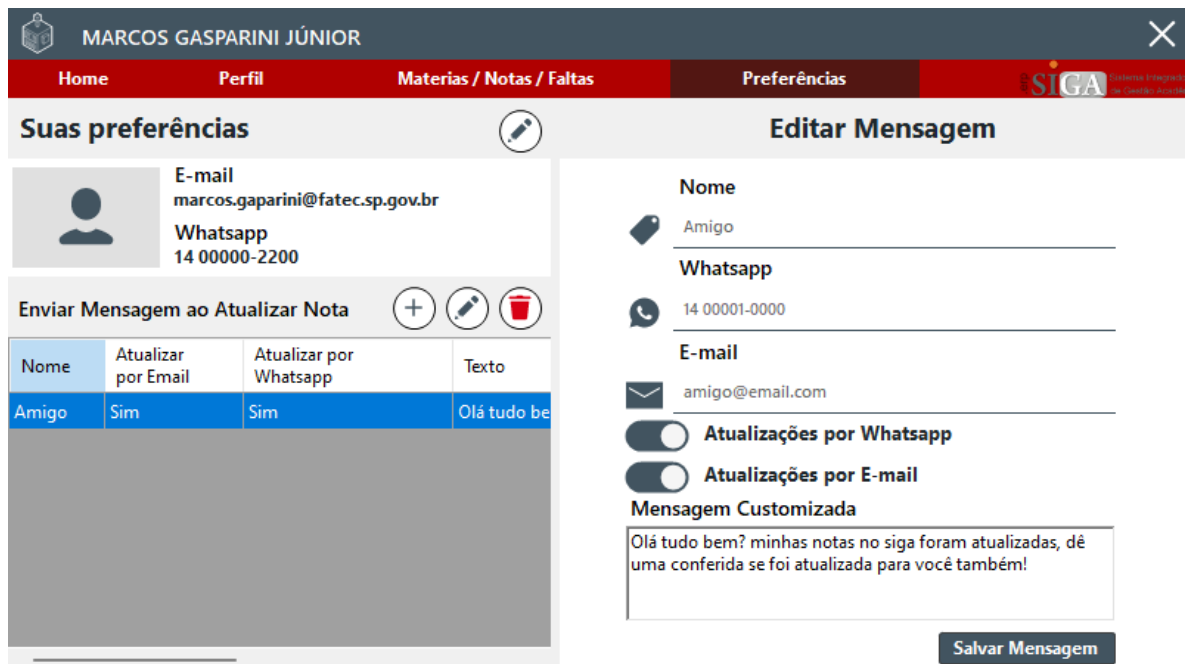


Figura 14 – Tela de Primeiro Acesso



Figura 15 – Tela de Primeiro Acesso

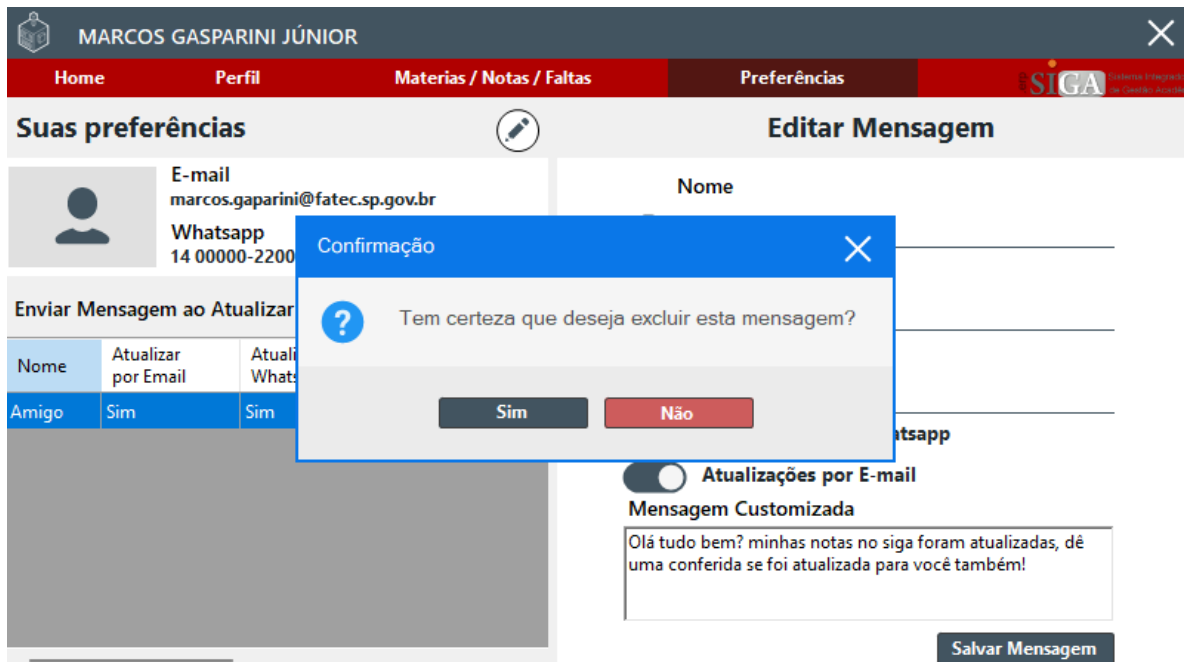
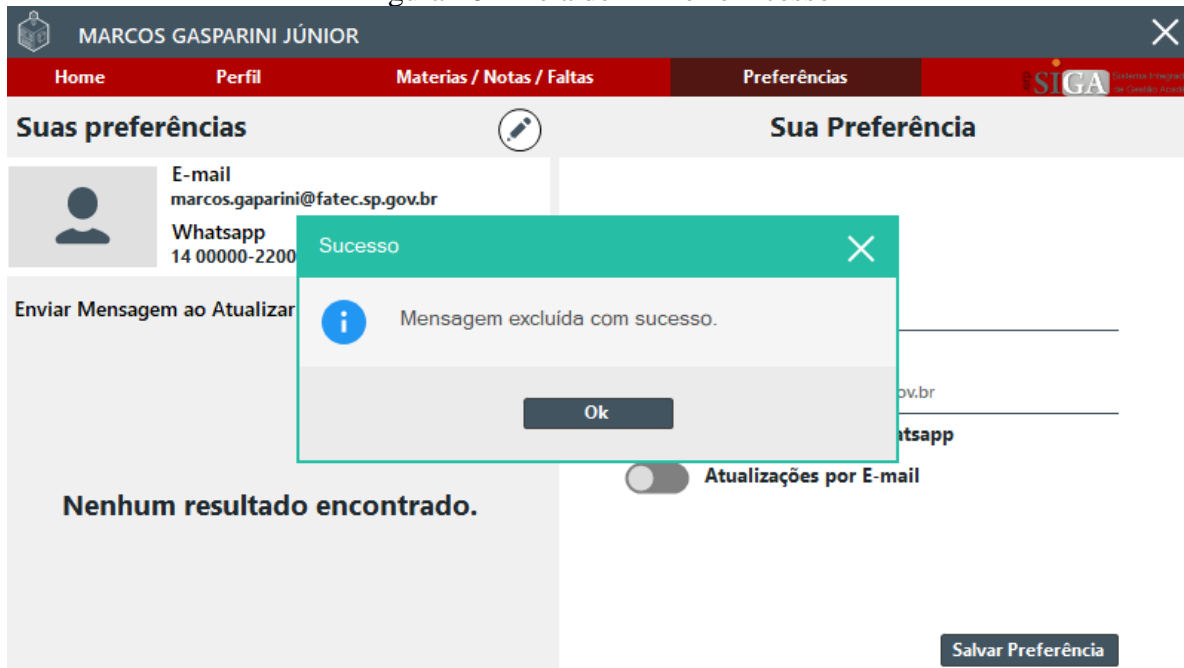


Figura 16 – Tela de Primeiro Acesso



A tela de Preferências, ilustrada na Figura 12, oferece aos usuários a possibilidade de controlar quem e de que maneira serão notificadas as alterações no sistema:

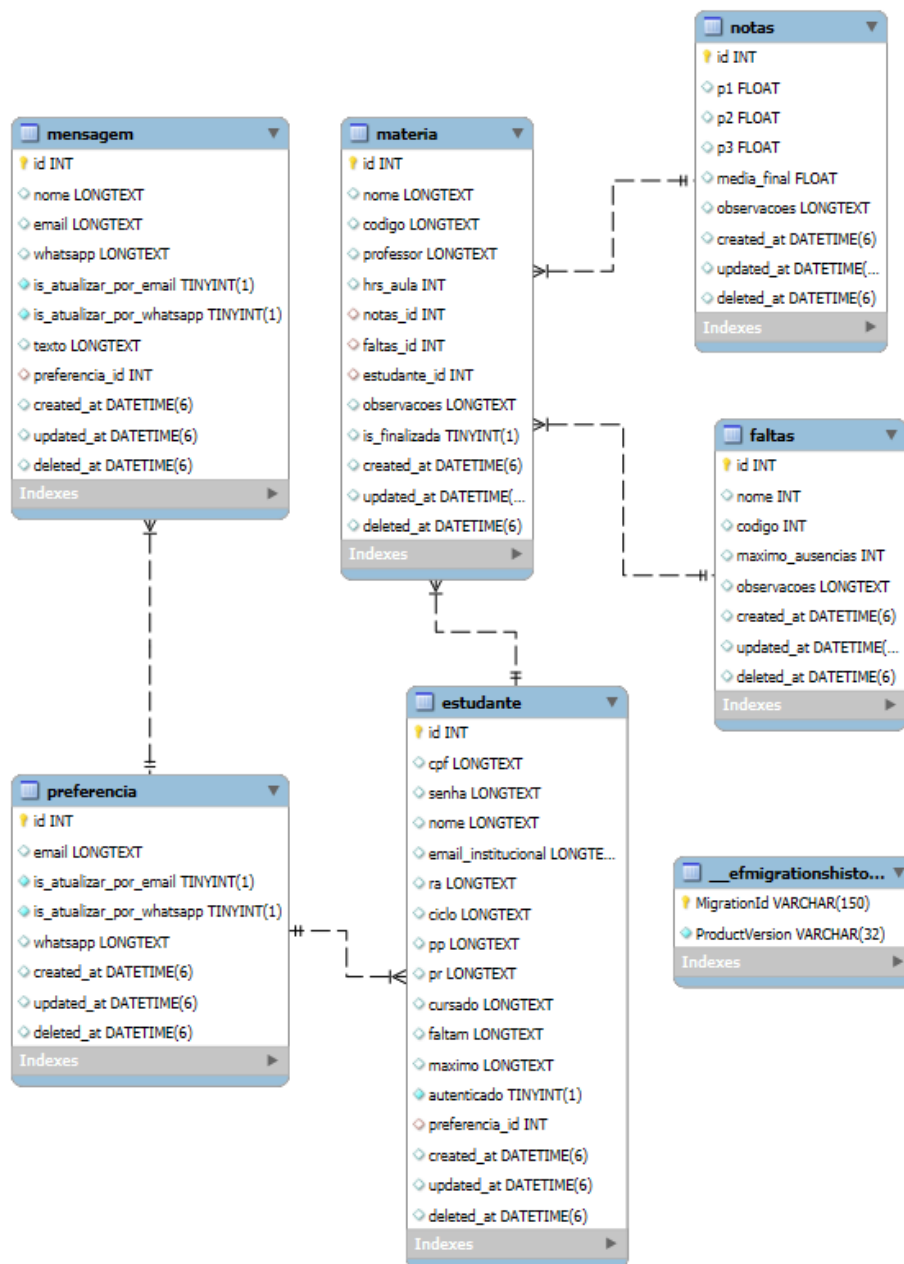
- Área "Suas Preferências": Campo para inserir a matéria pela qual o usuário deseja realizar a busca no sistema.
- Botão Editar Suas Preferências: Prepara a tela para editar as preferências de envio de mensagens do usuário.
- Botão Adicionar Mensagem: Prepara a tela para adicionar uma nova mensagem ao sistema.
- Botão Editar Mensagem: Permite editar uma mensagem já cadastrada no sistema.
- Botão Excluir Mensagem: Exclui uma mensagem cadastrada no sistema, exibindo uma confirmação conforme mostrado nas Figuras 15 e 16.
- Botão Salvar Preferências: Aparece apenas ao editar uma preferência (Figura 12), permitindo salvar as configurações de envio de mensagens para o usuário logado.
- Botão Salvar Mensagem: Aparece apenas ao editar uma mensagem (Figura 13), salvando as configurações de envio da mensagem para outros destinatários.
- Campo Nome: Campo para inserir o nome do destinatário da mensagem em caso de alterações nos dados.
- Campo WhatsApp: Configura o número de WhatsApp do destinatário para envio de mensagens quando forem identificadas alterações nos dados.
- Campo E-mail: Configura o E-mail do destinatário para notificações de alterações.



- Seleção Atualizar por WhatsApp: Habilita o envio de notificações via WhatsApp.
- Seleção Atualizar por E-mail: Habilita o envio de notificações via E-mail.
- Mensagem Customizada: Campo exibido apenas para envio de mensagens a terceiros, onde o usuário pode inserir o texto que será enviado.

#### 4.41 Diagrama Banco de Dados

Figura 50 – Diagrama Banco de Dados



## **5 APLICABILIDADE**

A aplicação é especialmente direcionada para líderes de sala, coordenadores ou representantes estudantis que buscam auxiliar seus colegas de classe, mantendo-os atualizados sobre as mudanças acadêmicas. Ao automatizar o monitoramento da plataforma SIGA, esses líderes podem receber notificações imediatas sobre atualizações, como a publicação de notas, e repassá-las rapidamente aos estudantes, garantindo uma comunicação interna mais eficiente.

Utilizando o WhatsApp Web JS, a ferramenta envia notificações diretamente aos contatos autorizados, permitindo uma comunicação rápida e eficiente por meio do aplicativo de mensagens mais utilizado. Esse mecanismo garante que os alunos ou outros usuários do sistema sejam informados em tempo real sobre as mudanças nas suas informações acadêmicas, sem precisar acessar constantemente a plataforma.

O objetivo do sistema é proporcionar uma solução simplificada e eficiente para a automação do processo de monitoramento de plataformas acadêmicas, contribuindo para a inclusão digital e facilitando o acesso e o gerenciamento das informações. Seu foco está em criar um ambiente mais dinâmico e funcional para o gerenciamento de atividades acadêmicas, oferecendo uma ferramenta ágil para que líderes de sala possam notificar seus colegas sobre as alterações importantes na plataforma.

## REFERÊNCIAS

**AMAZON.** *O que é uma API (interface de programação de aplicações)?*. 2024. Disponível em: <https://aws.amazon.com/pt/whatis/api/#:~:text=di%C3%A1rias%20no%20telefone,-,O%20que%20significa%20API%3F,de%20servi%C3%A7o%20entre%20duas%20aplica%C3%A7%C3%B5es.> . Acesso em: 5 nov. 2024.

**EXPRESS.** *Framework web rápido, flexível e minimalista para*. 2024. Disponível em: <https://expressjs.com/pt-br/> . Acesso em: 5 nov. 2024.

**MICROSOFT.** *Hub de documentação do Entity Framework*. 2024. Disponível em: <https://learn.microsoft.com/pt-br/ef/>. Acesso em: 5 nov. 2024.

**MICROSOFT.** *Guia da área de trabalho (Windows Forms .NET)*. 2024. Disponível em: [https:// https://learn.microsoft.com/pt-br/dotnet/desktop/winforms/overview/?view=netdesktop-8.0](https://learn.microsoft.com/pt-br/dotnet/desktop/winforms/overview/?view=netdesktop-8.0). Acesso em: 5 nov. 2024.

**SELENIUM.** *The Selenium Browser Automation Project*. 2024. Disponível em: <https://www.selenium.dev/documentation/> . Acesso em: 5 nov. 2024.

**SIGA.** *Sistema Integrado de Gestão Acadêmica*. 2024. Disponível em: <https://siga.cps.sp.gov.br/aluno/home.aspx>. Acesso em: 5 nov. 2024.

**WHATSAPPWEBJS.** *whatsapp-web.js 1.26.1-alpha.2*. 2024. Disponível em: <https://whatsapp-web.js 1.26.1-alpha.2> . Acesso em: 5 nov. 2024.

LINKS para o código fonte:

<https://github.com/JunynBoy/Bot.Siga>

<https://github.com/JunynBoy/whatsapp-web-js-api>