

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DO RIO GRANDE DO SUL  
CAMPUS RESTINGA  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**μIF: ARQUITETURA BASEADA EM MICROSERVIÇOS PARA  
GESTÃO ESCOLAR**

**JONAS AYRES DA SILVA**

**PORTO ALEGRE  
2021**

**JONAS AYRES DA SILVA**

**μIF: ARQUITETURA BASEADA EM MICROSERVIÇOS PARA GESTÃO  
ESCOLAR**

Trabalho de Conclusão de Curso apresentada como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas da Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - IFRS, Campus Restinga.

Orientador: Prof. Rafael Pereira Esteves

**JONAS AYRES DA SILVA**

**μIF: ARQUITETURA BASEADA EM MICROSERVIÇOS PARA GESTÃO  
ESCOLAR**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - IFRS, Campus Restinga.

Data de Aprovação:

**Banca Examinadora**

---

Prof. Rafael Pereira Esteves - IFRS - Campus Restinga  
Orientador

---

Prof. Roben Castagna Lunardi- IFRS- Campus Restinga  
Membro da Banca

---

Prof. Iuri Albandes Cunha Gomes- IFRS- Campus Restinga  
Membro da Banca

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL

Reitor: Prof. Júlio Xandro Heck

Pró-Reitor de Ensino: Prof. Lucas Coradini

Diretor do Campus Restinga: Prof. Rudinei Müller

Coordenador do CST em Análise e Desenvolvimento de Sistemas: Prof. Iuri Albandes Cunha Gomes

Bibliotecária-Chefe do Campus Restinga: Paula Porto Pedone

## RESUMO

Derivada da Arquitetura Orientada a Serviços e amplamente usada para substituir a arquitetura monolítica, a arquitetura de microsserviços é um dos assuntos mais emergentes na indústria e mais recentemente também na área acadêmica. Microsserviços podem ser desenvolvidos utilizando diferentes linguagens de programação e usar as tecnologias de banco de dados que melhor atendam as necessidades de cada serviço. Nesse contexto, o objetivo deste trabalho é a proposta de uma aplicação baseada em microsserviços destinada a automação e digitalização de formulários, solicitações e processos do IFRS. Como benefícios, espera-se obter facilidade no desenvolvimento de novos serviços e aumento na escalabilidade da aplicação. Como resultados, conclui-se que uma arquitetura baseada em microsserviços possui uma complexidade elevada se comparada a arquiteturas antecessoras, mas seus ganhos com facilidade de manutenção e novas implementações e heterogeneidade tecnológica tornam o desenvolvimento mais ágil e produtivo.

**Palavras-chave:** Microsserviços, Arquitetura de software, Desenvolvimento de software.

## **ABSTRACT**

Derived from Service Oriented Architecture and widely used to replace monolithic architecture, microservice architecture is one of the most emerging topics in the industry and more recently also in the academic field. Microservices can be developed using different programming languages and database technologies that best fit the needs of each service. In this context, the objective of this paper is to propose and develop an application based on microservices aimed at the automation and digitization of forms, requests, and IFRS processes. As benefits, it is expected to obtain efficiency in the development of new services and increase the scalability of the application. As a result, it is concluded that a microservices-based architecture has a high complexity when compared to predecessor architectures, but with ease of maintenance, ease of new features and technological heterogeneity make the development considerably more agile and productive.

**Keywords:** Microservices, Software architecture, Software development.

## LISTA DE FIGURAS

Figura 1 – Arquitetura	monolítica			
Fonte: Autor		14		
Figura 2 – Escalabilidade	e	complexidade		
Fonte: Autor		18		
Figura 3 – Visão	geral	da	aplicação	
Fonte: Autor				21
Figura 4 – Exemplo de documento <i>JSON</i> persistido na plataforma <i>MongoDB Atlas</i>				
Fonte: Autor				25
Figura 5 – Exemplo de código <i>React.js</i> + <i>Material-UI</i>				
Fonte: Autor				26
Figura 6 – Projeto “TCC” e seus respectivos repositórios hospedados no <i>GitHub</i>				
Fonte: Autor				27
Figura 7 – Serviços hospedados na plataforma <i>Heroku</i>				
Fonte: Autor				27
Figura 8 – Seção do código responsável pela formatação e envio de e-mails				
Fonte: Autor				28
Figura 9 – Tela	de	<i>login</i>		
Fonte: Autor				29
Figura 10 – Cabeçalho	com	opção	de	<i>Logout</i>
Fonte: Autor				29
Figura 11 – Listagem	de	solicitações		
Fonte: Autor				30
Figura 12 – Formulário	de	trancamento	de	matrícula
Fonte: Autor				31
Figura 13 – Seção de deferimento de solicitação de trancamento de matrícula				
Fonte: Autor				31
Figura 14 – Formulário	de	trancamento	de	disciplinas
Fonte: Autor				32
Figura 15 – Seção de deferimento de solicitação de trancamento de disciplinas				
Fonte: Autor				33
Figura 16 – Solicitação	de	justificativa	de	faltas
Fonte: Autor				33
Figura 17 – Seção de deferimento de justificativa de faltas				
Fonte: Autor				34

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TÉORICA</b>	<b>16</b>
2.1	Arquitetura orientada a serviços	16
2.2	Arquitetura de microsserviços	17
2.3	Virtualização baseada em <i>containers</i>	19
<b>3</b>	<b>SOLUÇÃO CONCEITUAL</b>	<b>20</b>
3.1	Visão geral	20
3.2	Serviços	21
3.2.1	Serviço de trancamento de matrícula	21
3.2.2	Serviço de trancamento de disciplina	22
3.2.3	Serviço de justificativa de faltas	23
3.2.4	Serviço de envio de e-mails	24
<b>4</b>	<b>IMPLEMENTAÇÃO</b>	<b>25</b>
4.1	Tecnologias utilizadas	25
4.1.1	<i>MongoDB</i>	25
4.1.2	<i>Node.js</i>	26
4.1.3	<i>React.js</i> e <i>Material-UI</i>	26
4.1.4	<i>GitHub</i>	26
4.1.5	<i>Heroku</i>	27
4.1.6	<i>FireBase</i>	27
4.1.7	<i>MailJet</i>	27
4.2	Funcionalidades	28
4.2.1	<i>Login</i>	28
4.2.2	<i>Logout</i>	29
4.2.3	Listagem de solicitações	29
4.2.4	Solicitação de trancamento de matrícula	30
4.2.5	Deferimento de trancamento de matrícula	31
4.2.6	Solicitação de trancamento de disciplinas	32
4.2.7	Deferimento de trancamento de disciplinas	32
4.2.8	Solicitação de justificativa de faltas	33
4.2.9	Deferimento de justificativa de faltas	33

<b>5</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>35</b>
5.1	CONCLUSÃO . . . . .	35
5.2	TRABALHOS FUTUROS . . . . .	35
	<b>REFERÊNCIAS . . . . .</b>	<b>37</b>

## 1 INTRODUÇÃO

A chamada “arquitetura monolítica” é utilizada para descrever aplicações de um único executável, no qual toda a modularização utilizada é executada em uma mesma máquina ou servidor, formando assim um “bloco” (ou monolito). Desta forma, os módulos e funcionalidades compartilham recursos de processamento, memória, bancos de dados e arquivos. (MACHADO, 2021)

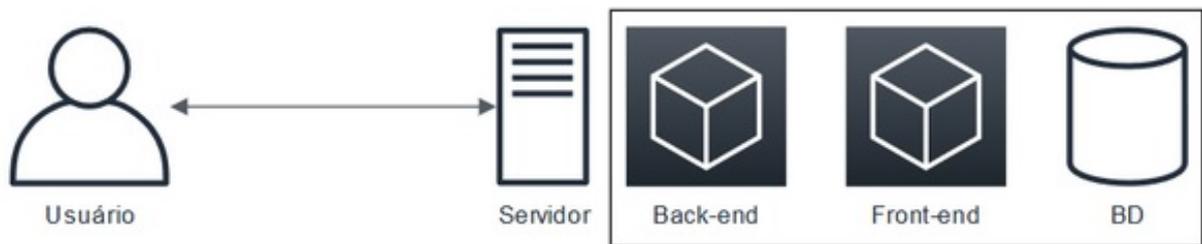


Figura 1 – Arquitetura monolítica

Fonte: Autor

Uma arquitetura monolítica típica de um sistema complexo pode ser representada pela figura acima, na qual todas as funções do negócio estão implementadas em um único processo (MACHADO, 2021). Atualmente, esse padrão de arquitetura é consolidado na indústria, sendo amplamente utilizado para resolução e aplicação de inúmeras soluções. Dois dos maiores problemas relacionados a esse padrão correspondem a dificuldade de escalabilidade e manutenção dessas aplicações, ou seja, uma vez que uma das funcionalidades da aplicação possui um mal funcionamento ou precisa de uma atualização, todas as funcionalidades ficam temporariamente indisponíveis. Uma solução adotada para esses problemas foi a implementação de aplicações onde o banco de dados seria totalmente independente da aplicação, possibilitando a chamada “escalabilidade vertical”, onde é necessário aumentar a capacidade de processamento do servidor onde a aplicação se encontra. Após a implementação desse padrão de arquitetura, o mesmo foi ampliado para a camada de apresentação dos sistemas monolíticos, fazendo com que essa camada se comunique com outra aplicação responsável pela camada de regras de negócio, removendo a dependência de ambas as partes, aumentando a escalabilidade de cada uma. Tais padrões deram origem a arquitetura baseada em microsserviços. Segundo Martin Fowler e James Lewis, arquitetura de microsserviços é uma abordagem que consiste em desenvolver uma única aplicação composta por pequenos serviços (geralmente APIs) que se comunicam através de mecanismos leves (como chamadas HTTP). Esses serviços são construídos se agrupando

em responsabilidades de negócio e implantados independentemente e de forma automatizada, podendo ser desenvolvidos usando diferentes linguagens e fazendo uso de diferentes tecnologias de banco de dados (LEWIS; FOWLER, 2019).

A arquitetura de microsserviços pode ser utilizada na construção de aplicações com as mais diversas finalidades. Neste trabalho, o foco é o desenvolvimento de sistemas para apoiar processos relacionados a gestão escolar, especificamente os que fazem parte da rotina do IFRS *campus* Restinga.

Atualmente, alguns processos de solicitações do IFRS *campus* Restinga, como por exemplo as de trancamento de disciplinas e de matrículas, são feitos de forma manual desde seu requerimento pelo aluno até a chegada ao coordenador do curso específico. Tendo como contexto a automação e digitalização de tais processos, esse Trabalho de Conclusão de Curso focará na proposta e desenvolvimento de uma aplicação utilizando o padrão de microsserviços visando um melhor desempenho, menor custo de desenvolvimento e consumo de recursos computacionais.

O objetivo principal deste trabalho é o desenvolvimento de uma aplicação de microsserviços destinada a automação e digitalização de formulários, requerimentos e processos do IFRS - *campus* Restinga. Como benefícios da adoção desse padrão, espera-se obter facilidade no desenvolvimento de novos serviços conforme a necessidade do *campus* e aumento na escalabilidade para suportar um grande número de requisições, especialmente em períodos de pico.

Além deste capítulo introdutório, o trabalho está organizado da seguinte forma. O Capítulo 2 detalha o conceito de microsserviços, apresentado as vantagens desse padrão em relação ao modelo monolítico. O Capítulo 3 apresenta o estudo de caso de implementação e definição de padrões e arquitetura de microsserviços a ser utilizado no contexto dos processos do IFRS *campus* Restinga. O capítulo 4 mostra detalhes de desenvolvimento, ferramentas e tecnologias que foram utilizadas. O capítulo 5 conclui o trabalho e apresenta possibilidades de trabalhos futuros e melhorias.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 ARQUITETURA ORIENTADA A SERVIÇOS

Computação Orientada a Serviços (Service-Oriented Computing - SOC), tem por objetivo atacar a complexidade dos sistemas distribuídos e integrar diferentes aplicações de software (MACKENZIE et al., 2006). No modelo SOC, uma aplicação é chamada de serviço e cada serviço oferece funcionalidades que se comunicam por meio do envio de mensagens. As principais características do SOC são: dinamismo, modularidade, reuso, desenvolvimento distribuído e integração heterogênea de sistemas. Segundo Dragoni et al., no início dos anos 2000 surgia a primeira geração de serviços baseada no paradigma de SOC, a Arquitetura Orientada a Serviços (SOA). Essa arquitetura utilizou os mesmos conceitos de SOC, porém adicionou a definição de capacidade de negócio, o que permitiu integrar diferentes serviços de negócios. (DRAGONI et al., 2017)

A arquitetura SOA é um tipo de arquitetura de software que permite que um sistema distribuído utilize funcionalidades implementadas por diferentes equipes ou providas por terceiros, independente da tecnologia adotada. Josuttis define SOA como um paradigma, ou conjunto de princípios, que guia a concepção e manutenção de processos de negócio inerentes a grandes sistemas distribuídos. Desta forma, SOA não é uma solução pronta para ser implantada, mas algo que deve ser elaborado de acordo com o contexto e os requisitos de cada situação. Os princípios de SOA são baseados nos seguintes aspectos (JOSUTTIS, 2007):

- Serviço-funcionalidade autocontida: simples ou complexa, que utiliza interfaces para conectar as necessidades do negócio à tecnologia de informação. O uso de interfaces formalmente definidas permite a interoperabilidade de forma heterogênea e encapsulada, independentemente da linguagem adotada;
- *Enterprise Service Bus (ESB)*: é a infraestrutura que torna capaz a interoperabilidade entre os serviços, independentemente de plataformas e tecnologias;
- Baixo Acoplamento: preza pela redução das dependências do sistema, resultando na redução da superfície de contato para efeitos de modificações, aumento na tolerância a falhas, escalabilidade e flexibilidade.

Para adotar uma arquitetura de sistemas distribuídos, um projeto precisa ter um conjunto bem definido de papéis, políticas e processos em todas as equipes que irão colaborar na entrega da

solução final integrada. Esses conjuntos serão as ferramentas que conduzirão uma implementação bem sucedida da SOA.

## 2.2 ARQUITETURA DE MICROSERVIÇOS

Conforme Dragoni et al., microsserviços é a segunda geração de serviços baseada em SOA. Trata-se de uma abordagem que utiliza conceitos bases do SOC, como a separação de responsabilidades, e remove as complexidades geradas pelo SOA, como utilização de ESB e orquestração de serviços centrada no processo de negócio, focando somente na implementação de serviços simples com uma única responsabilidade. Dessa forma, microsserviços é considerada uma abordagem simplificada para construção de sistemas distribuídos. (DRAGONI et al., 2017)

Fowler define que o estilo arquitetural dos microsserviços aborda como desenvolver uma única aplicação como um conjunto de pequenos serviços, cada um executando em seu próprio processo e utilizando mecanismos leves de comunicação. Estes serviços são construídos em torno de competências de negócio e são disponibilizados independentemente por processos de implantação automatizados. Há um gerenciamento centralizado mínimo desses serviços, que podem ser desenvolvidos em diferentes linguagens de programação e utilizar distintas tecnologias de armazenamento de dados. (FOWLER, 2014)

Nos últimos anos, vem crescendo o interesse tanto na indústria como na academia por esse tema. Conforme Nadareishvili et al., um dos principais interesses ao adotar a implementação de microsserviços está relacionado ao equilíbrio entre velocidade e segurança no desenvolvimento (NADAREISHVILI et al., 2016). Segundo os autores, aplicações tendem a se tornar mais complexas a medida que sua escala na forma de escopo, volume e interações do usuário aumenta. Para eles, a arquitetura de microsserviços está sendo utilizada para realizar entregas mais rápidas e ter maior segurança ao aumentar a escala de um sistema, como ilustrado na imagem abaixo.

Newman cita sete benefícios na utilização de microsserviços: heterogeneidade tecnológica, resiliência, escalabilidade, fácil implantação, alinhamento organizacional, modularidade e otimização de substituição. (NEWMAN, 2014)

- Heterogeneidade Tecnológica: a autonomia que os microsserviços trazem permite que as equipes de desenvolvimento adotem uma nova tecnologia de forma mais frequente e rápida. A independência de microsserviços permite que as equipes tenham mais liberdade de definição das tecnologias de que eles necessitam para solucionar um determinado problema.

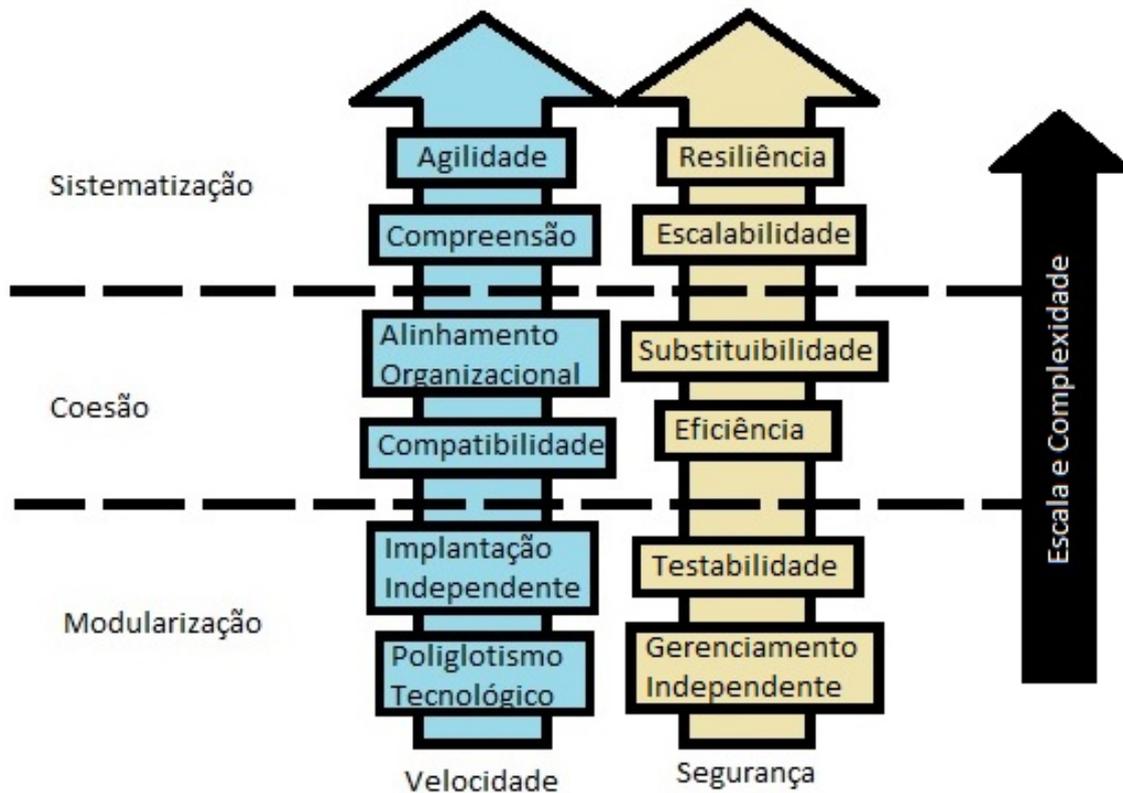


Figura 2 – Escalabilidade e complexidade

Fonte: Autor

- **Resiliência:** uma das certezas existentes no desenvolvimento de software é que em determinado momento, tanto a rede quanto a máquina poderão falhar. A arquitetura de microsserviços tende a ser tolerante a falhas devido a sua característica de independência. Cada serviço poderá tratar de forma isolada tais (e outras) interrupções, evitando assim a parada momentânea do sistema como um todo, como ocorre em sistemas monolíticos.
- **Escalabilidade:** um sistema de microsserviços permite uma escalabilidade de forma granular, ou seja, se um microsserviço necessitar de mais recursos computacionais, como memória ou CPU, somente esse serviço será escalado e não todos os componentes como observado em aplicações monolíticas.
- **Fácil Implantação:** por serem independentes e autônomos, microsserviços podem ser implantados de forma mais simples. Tal característica de autonomia permite que uma aplicação tenha menor tempo de indisponibilidade ao realizar uma mudança de funcionalidade. Cada microsserviço sendo responsável por uma única funcionalidade, o gerenciamento de versões se torna menos oneroso, pois caso a nova versão não esteja com o funcionamento esperado, somente o serviço responsável por esse funcionamento precisa ser corrigido.

- Alinhamento organizacional: devido a suas características de autonomia e delimitação de contexto, um microsserviço tende a possuir bases de código menores e menos complexas. Permitindo a criação de times de desenvolvimento mais focados em um determinado contexto de negócio e com uma maior produtividade.
- Modularidade: não sendo limitados a determinada tecnologia, cada microsserviço pode ser consumido de diferentes formas e para diferentes propósitos por qualquer tipo de aplicação.
- Otimização a substituição: por serem simples e autônomos, microsserviços permitem a re-escrita e remoção de funcionalidades de forma facilitada, encorajando assim a manutenção de sua base de código.

### 2.3 VIRTUALIZAÇÃO BASEADA EM *CONTAINERS*

*Container* é um método de virtualização de sistemas operacionais que permite executar uma aplicação, suas configurações e suas dependências em processos com recursos isolados oferecendo um ambiente consistente e produtivo ao desenvolvimento e ao controle de versão. Segundo Agarwal, virtualização a nível de sistemas operacionais é a existência de múltiplas instâncias de espaço de usuários isolados (AGARWAL, 2015). Essas instâncias tem seus próprios recursos, sistema de arquivos e são executadas completamente isoladas uma das outras. Com a criação de *containers* a aplicação se torna abstraída de todas as infraestruturas subjacentes, sendo essencial para o desenvolvimento isolado de serviços em uma arquitetura de microsserviços. Em comparação com as máquinas virtuais, os *containers* se tornam mais leves, pois as aplicações que fazem uso dos mesmos utilizam chamadas ao sistema operacional *host*, ou seja, todos os *containers* utilizam um único sistema operacional e compartilham o *kernel* do sistema entre si, os tornando mais leve e consumindo menos recursos que as máquinas virtuais.

Atualmente sendo a ferramenta mais popular para a facilitação e configuração do uso de *containers*, *Docker* possibilita a gestão de toda infraestrutura de uma aplicação reproduzindo o processo de empacotamento dentro de *containers*. De forma prática, a tecnologia possibilita a portabilidade de ambientes inteiros de forma ágil e simplificada, reduzindo o tempo de implantação e configuração de aplicações, dessa forma sendo a tecnologia mais indicada para a virtualização e implantação de aplicações baseadas em microsserviços.

### 3 SOLUÇÃO CONCEITUAL

Os estudantes do IFRS - *campus* Restinga podem realizar várias solicitações relacionadas a sua vida estudantil. Este trabalho foca nas seguintes solicitações:

- Trancamento de matrícula: o estudante interrompe formalmente o curso com a intenção de retomar de onde parou, sem perder o vínculo com a instituição. Essa solicitação é semestral;
- Trancamento de disciplina: diferente do trancamento de matrícula, nessa solicitação o aluno interrompe somente uma disciplina que o mesmo esteja cursando. Também funciona de forma semestral;
- Justificativa de faltas: o aluno que não tiver no mínimo 75% de frequência às aulas reprovará automaticamente na disciplina, independente de eventual aprovação por notas e/ou avaliação. Nesse contexto, por meio dessa solicitação o aluno pode abonar faltas consideradas justificadas pela instituição.

Os processos citados acima possuem a necessidade de comparecimento do estudante à instituição e também a espera da análise e possível deferimento por parte do coordenador do curso. Esse processo ocorre de forma lenta e é passível de erros durante sua análise.

Nesse contexto, o objetivo deste trabalho é desenvolver uma solução baseada em micro-serviços com o intuito de automatizar esses processos.

#### 3.1 VISÃO GERAL

Uma visão geral da aplicação é ilustrada na Figura 3.1

Atualmente a suíte de serviços possui três aplicações de escopos fechados (uma para cada solicitação mencionada anteriormente) e uma aplicação utilitária de envio de e-mails, todas elas sendo totalmente independentes desde seu banco de dados até sua implantação. Atualmente, o banco de dados de cada serviço tem como finalidade a criação de histórico de seu respectivo requerimento.

A aplicação funciona da seguinte forma: Com uma das permissões existentes (sendo elas administrador, aluno, professor e coordenador), o usuário realiza a autenticação necessária para ter acesso à interface da aplicação. Como aluno e administrador, o usuário terá acesso e poderá realizar todos os requerimentos disponíveis no sistema. Como administrador, professor e coordenador, o usuário terá acesso ao *dashboard* de deferimento de requerimentos pendentes.

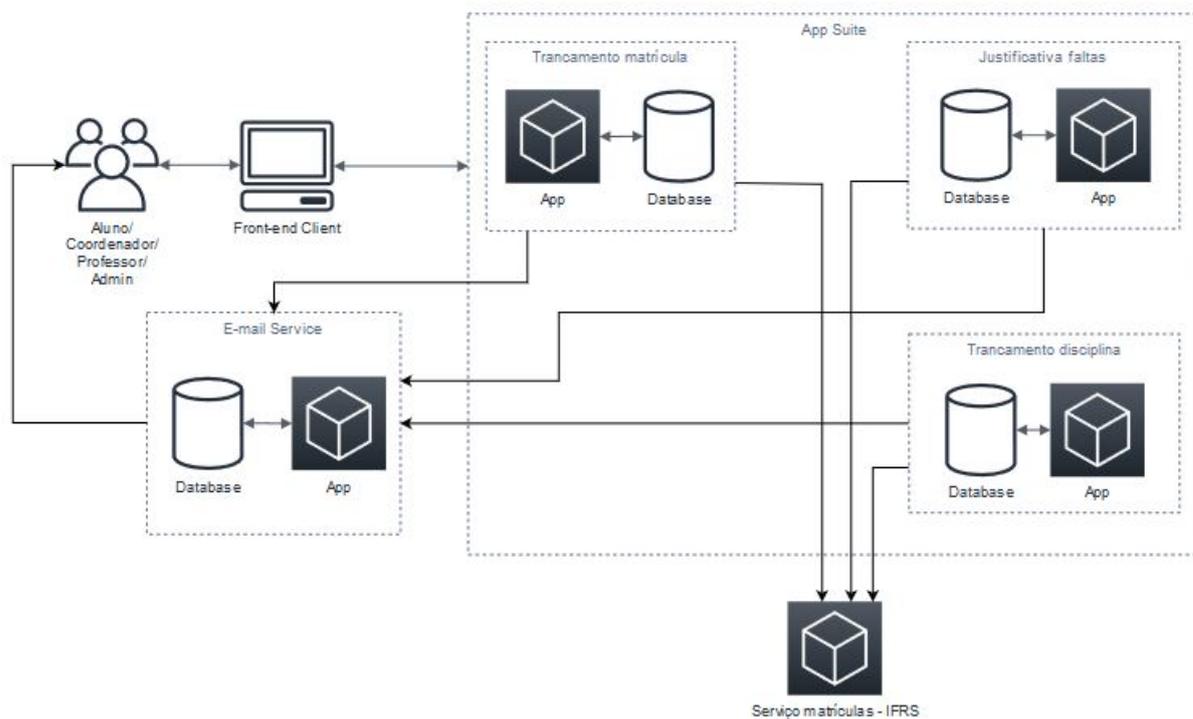


Figura 3 – Visão geral da aplicação  
Fonte: Autor

## 3.2 SERVIÇOS

### 3.2.1 Serviço de trancamento de matrícula

Esse serviço possui duas funcionalidades, sendo elas:

- **Solicitação de trancamento:** através de uma interface de formulário, o usuário (ou aluno) preenche os dados necessários para a solicitação, sendo eles: nome completo (campo textual), número de matrícula (campo numérico), ano/semestre de ingresso (campo numérico), curso (campo *select* ou *drop down*, com opções de curso fornecidas pela instituição), CPF (campo textual), data de encaminhamento (campo de data), se o aluno recebe assistência estudantil (*checkbox* - sim ou não), se o aluno possui débitos na biblioteca (*checkbox* - sim ou não), se o aluno possui licença maternidade (*checkbox* - sim ou não) e se o aluno possui comprovante de alistamento militar (*checkbox* - sim ou não). Ao submeter o formulário, o mesmo passa por uma série de validações:

1. se algum dos campos está vazio, sendo que todos são obrigatórios;
2. se a matrícula inserida existe nos registros da instituição e se a matrícula do aluno está ativa no momento da solicitação. Essa validação é feita por meio de comunicação

com o serviço de gerenciamento de dados da instituição;

3. se o motivo da solicitação for alistamento militar ou licença maternidade, o sistema valida a presença de documentos que comprovem os tais motivos.

Após as validações, a solicitação é persistida no banco de dados e um e-mail é enviado para o coordenador do curso correspondente a solicitação. Esse e-mail é enviado por meio de uma comunicação com a aplicação de envio de e-mails.

- **Deferimento de solicitação de trancamento:** No papel de coordenador do curso do trancamento solicitado, como mencionado anteriormente, o usuário recebe um e-mail quando uma nova solicitação é cadastrada no serviço. Este e-mail possui um *link* que redirecionará o usuário para uma página do sistema onde ele terá os dados da solicitação para fins de análise por parte do coordenador, e também duas opções de submissão de formulário: “Deferir” e “Indeferir”. Independente da opção escolhida pelo usuário, a aplicação receberá o deferimento e persistirá no banco de dados e enviará um e-mail para o solicitante, notificando o mesmo sobre a situação de sua solicitação.

### 3.2.2 Serviço de trancamento de disciplina

Esse serviço possui duas funcionalidades, sendo elas:

- **Solicitação de trancamento:** através de uma interface de formulário, o usuário (ou aluno) preenche os dados necessários para a solicitação, sendo eles: nome completo (campo textual), número de matrícula (campo numérico), ano/semestre de ingresso (campo numérico), curso (campo *select* ou *drop down*, com opções de curso fornecidas pela instituição), CPF (campo textual), data de encaminhamento (campo de data) e lista de disciplinas (campo *select* ou *drop down*, com opções de disciplinas que o aluno se encontra matriculado). Ao submeter o formulário, o mesmo passa por uma série de validações:

1. se algum dos campos está vazio, sendo que todos são obrigatórios;
2. se a matrícula inserida existe nos registros da instituição e se a matrícula do aluno está ativa no momento da solicitação. Essa validação é feita por meio de comunicação com o serviço de gerenciamento de dados da instituição.

Após as validações, a solicitação é persistida no banco de dados e um e-mail é enviado para o coordenador do curso correspondente a solicitação. Esse e-mail é enviado por meio de uma comunicação com a aplicação de envio de e-mails.

- **Deferimento de solicitação de trancamento:** No papel de coordenador do curso do trancamento solicitado, como mencionado anteriormente, o usuário recebe um e-mail quando uma nova solicitação é cadastrada no serviço. Este e-mail possui um *link* que redirecionará o usuário para uma página do sistema onde ele terá os dados da solicitação para fins de análise por parte do coordenador, e também duas opções de submissão de formulário: “Deferir” e “Indeferir”. Independente da opção escolhida pelo usuário, a aplicação receberá o deferimento e persistirá no banco de dados e enviará um e-mail para o solicitante, notificando o mesmo sobre a situação de sua solicitação.

### 3.2.3 Serviço de justificativa de faltas

Esse serviço possui três funcionalidades, sendo elas:

- **Solicitação de justificativa:** através de uma interface de formulário, o usuário (ou aluno) preenche os dados necessários para a solicitação, sendo eles: nome completo (campo textual), número de matrícula (campo numérico), ano/semestre de ingresso (campo numérico), curso (campo *select* ou *drop down*, com opções de curso fornecidas pela instituição), data de encaminhamento (campo de data) e lista de justificativas, que por sua vez possuem os campos: data da falta (campo de data), lista de disciplinas correspondentes a data da falta (campo *select* ou *drop down*, com opções de disciplinas que o aluno se encontra matriculado), professor (campo textual), se o aluno precisa de avaliação substitutiva (campo *checkbox* - sim ou não) e motivo (campo *select* ou *drop down*), que possui as seguintes opções: problema de saúde, serviço militar, falecimento de parente, convocação judiciária, convocação IFRS, atividade IFRS, autoridade militar, desporto IFRS, audiência judicial e demais casos. Ao submeter o formulário, o mesmo passa por uma série de validações:

1. se algum dos campos está vazio, sendo que todos são obrigatórios;
2. se a matrícula inserida existe nos registros da instituição e se a matrícula do aluno está ativa no momento da solicitação. Essa validação é feita por meio de comunicação com o serviço de gerenciamento de dados da instituição.

Após as validações, a solicitação é persistida no banco de dados e um e-mail é enviado para o coordenador do curso correspondente a solicitação. Esse e-mail é enviado por meio de uma comunicação com a aplicação de envio de e-mails.

- **Deferimento de solicitação de justificativa:** No papel de coordenador do curso da justificativa de faltas solicitada, como mencionado anteriormente, o usuário recebe um e-mail

quando uma nova solicitação é cadastrada no serviço. Este e-mail possui um *link* que redirecionará o usuário para uma página do sistema onde ele terá os dados da solicitação para fins de análise por parte do coordenador. Essa página terá as seguintes opções apresentadas: “Deferir”, “Indeferir” e caso conste na solicitação um pedido de avaliação substitutiva, também será apresentada uma opção para notificar o professor responsável pela disciplina em questão, enviando um e-mail para o mesmo. Independente da opção escolhida pelo usuário, a aplicação receberá o deferimento e persistirá no banco de dados e enviará um e-mail para o solicitante, notificando o mesmo sobre a situação de sua solicitação.

- **Envio de notificação a professores:** No papel de coordenador do curso, ao analisar uma solicitação de justificativa de faltas, se a solicitação possuir a necessidade de uma avaliação substitutiva, o usuário pode enviar um e-mail de notificação para o professor responsável pela disciplina em questão.

#### 3.2.4 Serviço de envio de e-mails

Pode ser usado por qualquer outro serviço. Esta aplicação não possui interface gráfica (somente *back-end*). Esse serviço recebe chamadas HTTP com os seguintes dados:

- “*content*”: sendo o conteúdo ou corpo do email, contendo a mensagem a ser enviada;
- “*to*”: sendo o endereço de e-mail que receberá a dita mensagem;
- “*subject*”: sendo o assunto do e-mail.

Após o recebimento destes dados, o serviço envia o e-mail recebido e persiste os dados do mesmo no banco de dados, para fins de auditoria.

## 4 IMPLEMENTAÇÃO

Abaixo serão listadas e apresentadas as tecnologias utilizadas e também um descritivo das funcionalidades da aplicação.

### 4.1 TECNOLOGIAS UTILIZADAS

As tecnologias utilizadas no desenvolvimento da aplicação foram escolhidas com base na relevância no mercado de trabalho, uso abrangente, facilidade de implementação e experiência de uso do autor. Entre as tecnologias utilizadas estão *Node.js*, *MongoDB* e *React.js* juntamente com *Material-UI*. Para o controle de versão e hospedagem do código-fonte foi utilizada a plataforma *GitHub*, *Heroku* foi utilizado para a implantação dos serviços, *Google FireBase* para autenticação de usuários e a plataforma *MailJet* para envio de e-mails, os três últimos sendo serviços de nuvem (*Cloud Services*).

#### 4.1.1 *MongoDB*

Sendo de código aberto e multiplataforma, é um banco de dados orientado a documentos, usando-os no formato *JSON* em seus esquemas. Se classifica como um banco de dados *NoSQL* (*Not Only SQL*, não necessariamente relacional) ou *NoREL* (*non-relational*, não relacional) (MEDEIROS, 2014), sendo a tecnologia mais utilizada dessa categoria, e o quinto banco de dados mais utilizado no geral. Foi escolhido, principalmente, devido a sua facilidade de migração ou atualização do modelo de dados a serem persistidos, fazendo desnecessárias novas implementações de configurações no banco de dados uma vez que atualizadas na aplicação. O plano gratuito do serviço de banco de dados em nuvem *MongoDB Atlas* foi utilizado como principal forma de persistência de dados da aplicação e seus serviços.

```

    _id: ObjectId("5fb588b36d3b13002432c418")
  anexos: Array
    nomeCompleto: "Fulano da Silva"
    matricula: 10070100
    anoSemestreIngresso: "2019/1"
    curso: "ADS"
  justificativas: Array
    0: Object
      _id: ObjectId("5fb588b36d3b13002432c419")
      dataDaFalta: 2020-10-10T00:00:00.000+00:00
      disciplinas: Array
        0: Object
          avaliacaoSubstitutiva: false
          _id: ObjectId("5fb588b36d3b13002432c41a")
          nome: "Redes I"
          professor: "Esteves"
          motivo: "demaisCasos"
          dataEncaminhamento: 2020-11-18T20:48:51.848+00:00

```

Figura 4 – Exemplo de documento *JSON* persistido na plataforma *MongoDB Atlas*  
Fonte: Autor

#### 4.1.2 *Node.js*

Utilizando a linguagem de programação *JavaScript (JS)*, *Node.js (Node)*, é um *framework* de código aberto e multiplataforma, sendo destinado para desenvolvimento de aplicações *back-end*. Sendo utilizado em todos os serviços do sistema como servidor HTTP, contendo as regras de negócio e validações gerais de cada aplicação. Entre as bibliotecas utilizadas no desenvolvimento se destacam:

- ***Express.js***: destinado para otimização e construção de servidores *web* e *APIs*;
- ***Jest.js***: utilizado para testes unitários;
- ***Mongoose.js***: utilizado para modelagem de dados e conexão com *MongoDB*.

#### 4.1.3 *React.js* e *Material-UI*

Também baseado em *JavaScript*, *React* é uma biblioteca de código aberto utilizada para desenvolver interfaces de usuário (*front-end*). *Material-UI* é uma biblioteca de componentes utilizados com *React*, desenvolvida com o intuito de facilitar e agilizar o desenvolvimento com a biblioteca.

```
<Grid item xs={4}>
  <FormControl>
    <TextField
      id='matricula'
      label='Matrícula'
      onChange={input => this.setState({ matricula: input.target.value })}
      value={this.state.matricula}
      inputProps={{ maxLength: 8 }}
    />
  </FormControl>
</Grid>
```

Figura 5 – Exemplo de código *React.js* + *Material-UI*

Fonte: Autor

#### 4.1.4 *GitHub*

Com 36 milhões de usuários ativos mundialmente e mais de 100 milhões de projetos, *GitHub* é uma plataforma de versionamento e hospedagem de arquivos. Ele permite que usuários cadastrados na plataforma contribuam em projetos privados e públicos (*Open Source*). Embora uma das maiores vantagens da plataforma ser o trabalho distribuído de um mesmo código, muitos

usuários utilizam o mesmo para projetos individuais devido à quantidade de recursos que ela oferece.

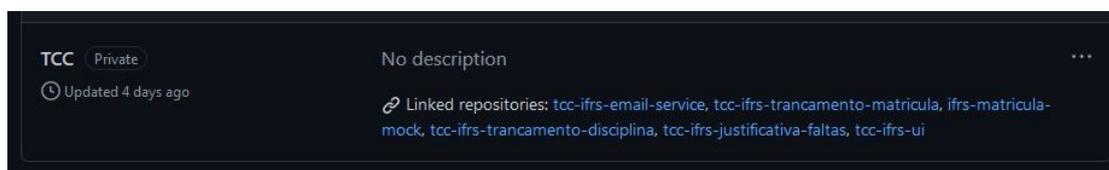


Figura 6 – Projeto “TCC” e seus respectivos repositórios hospedados no *GitHub*  
Fonte: Autor

#### 4.1.5 *Heroku*

Plataforma de nuvem como um serviço (*Cloud as a Service*) que suporta múltiplas linguagens de programação e tecnologias. Utilizado para a implantação de todos os serviços da aplicação, fazendo o gerenciamento de cada um dos mesmos de forma *containerizada* e automatizada, dispensando o uso de um arquivo de configuração.

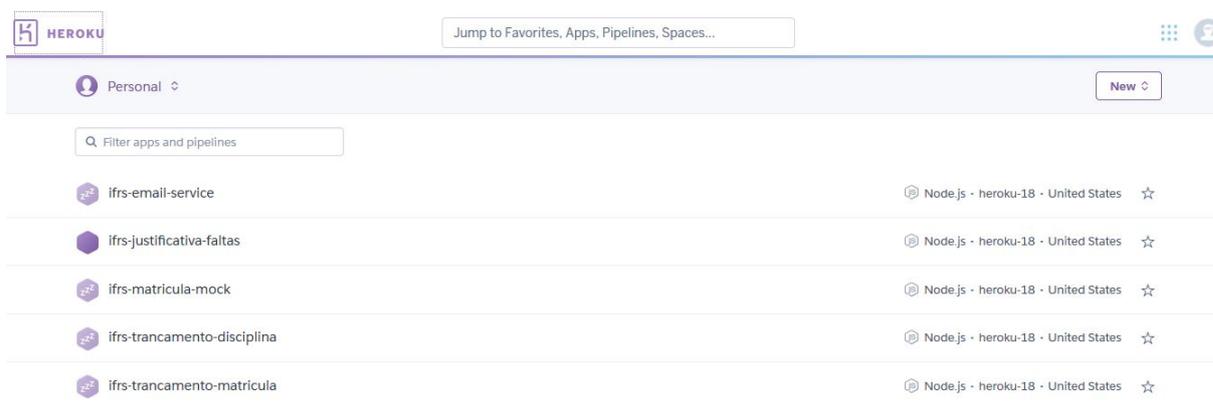


Figura 7 – Serviços hospedados na plataforma *Heroku*  
Fonte: Autor

#### 4.1.6 *FireBase*

Plataforma desenvolvida pela empresa *Google*. Originalmente servindo exclusivamente para autenticação de aplicações *web* e atualmente possuindo e oferecendo múltiplas opções de serviços. A plataforma foi utilizada na autenticação (*Login*) geral do sistema.

#### 4.1.7 *MailJet*

Plataforma de entrega e rastreamento de e-mails baseado em nuvem. Foi utilizada na aplicação para envio de notificações para usuários.

```
app.use('/api/sendEmail', (req, res) => {
  const body = req.body;

  mailjet.post('send', { 'version': 'v3.1' }).request({
    'Messages': [
      {
        'From': { 'Email': 'adm.temporario.ifrs@gmail.com' },
        'To': [ { 'Email': body.to } ],
        'Subject': body.subject,
        'TextPart': body.content,
        'CustomID': 'IFRSEmailSender'
      }
    ]
  }).then((result) => {
    res.status(200).send();
  }).catch((err) => {
    res.status(500).json({ ApiStatusCode: err.statusCode });
  });
});
```

Figura 8 – Seção do código responsável pela formatação e envio de e-mails

Fonte: Autor

## 4.2 FUNCIONALIDADES

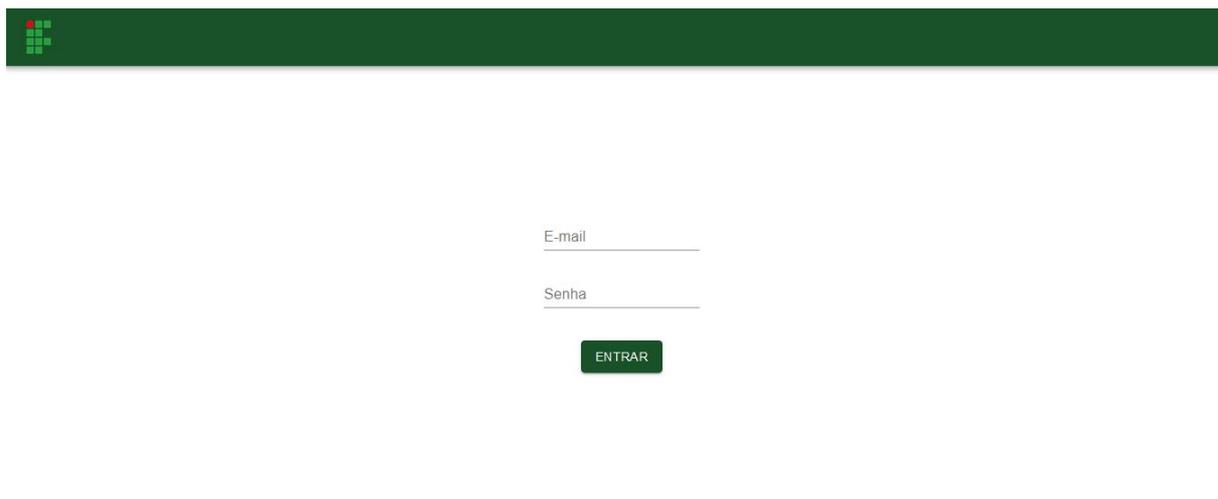
Esta seção descreve e ilustra as funcionalidades disponíveis na aplicação em formato similar a *User Stories*.

### 4.2.1 *Login*

Sendo aluno, coordenador ou administrador do sistema, desejo realizar autenticação para ter acesso à aplicação; Como acessar: não estando autenticado, o usuário será redirecionado a essa página sempre que tentar acessar qualquer seção do sistema;

Campos: E-mail e Senha;

Resultado: Ao clicar no botão "Entrar", com os dados corretos, o usuário será redirecionado a página inicial da aplicação.



A screenshot of a login page. At the top left, there is a small logo consisting of a grid of colored squares. Below it, there are two input fields: the first is labeled 'E-mail' and the second is labeled 'Senha'. Below these fields is a green button with the text 'ENTRAR' in white capital letters.

Figura 9 – Tela de *login*

Fonte: Autor

#### 4.2.2 *Logout*

Sendo aluno ou coordenador, desejo fazer *logout* para deixar de ter acesso à aplicação e proteger minha conta;

Como acessar: estando autenticado, o usuário deve clicar no botão "Sair" localizado no cabeçalho da página;

Resultado: Clicando sobre o botão, o usuário realiza o *logout* e é redirecionado para a tela de *login*.



Figura 10 – Cabeçalho com opção de *Logout*

Fonte: Autor

#### 4.2.3 Listagem de solicitações

Sendo coordenador, desejo visualizar todas as solicitações feitas para o curso que coordeno para então analisá-las;

Como acessar: estando autenticado, o usuário deve clicar na seção "Solicitações" localizada no cabeçalho da página, e então escolher uma das opções:

- Trancamento de disciplina;
- Trancamento de matrícula;

- Justificativa de faltas.

Resultado: Clicando sobre a seção, o usuário será redirecionado para a página de listagem, onde todas as solicitações do tipo escolhido são apresentadas.

A listagem possui os seguintes filtros de visualização:

- Sem filtro;
- Deferidos;
- Indeferidos;
- Não processados.

Solicitações de trancamento de matrícula				Filtrar
	Aluno	Matrícula	Curso	Data da solicitação
▼	Isabel Carvalho	10070101	ADS	19/02/2021
▼	Hugo Freire	10070100	ADS	19/02/2021
Itens por página: 5 1-2 de 2 < >				

Figura 11 – Listagem de solicitações

Fonte: Autor

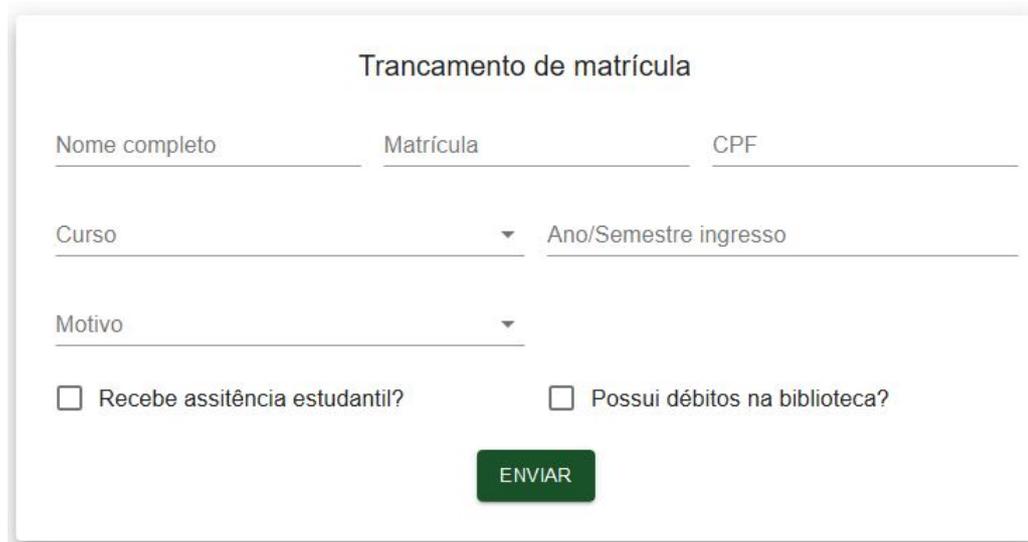
#### 4.2.4 Solicitação de trancamento de matrícula

Sendo aluno desejo realizar a requisição do trancamento de minha matrícula para assim não ter vínculo com o instituto por um semestre;

Como acessar: estando autenticado, o usuário deverá clicar no botão "Trancamento de matrícula";

Campos: Nome completo, número de matrícula, ano/semestre de ingresso, curso, CPF, débitos na biblioteca, licença maternidade e alistamento militar;

Resultado: Ao clicar no botão "Enviar", com os dados corretos, o formulário será reiniciado e um aviso será mostrado informando que o requerimento foi enviado e um e-mail será redigido ao coordenador do curso especificado.



**Trancamento de matrícula**

Nome completo \_\_\_\_\_ Matrícula \_\_\_\_\_ CPF \_\_\_\_\_

Curso \_\_\_\_\_ Ano/Semestre ingresso \_\_\_\_\_

Motivo \_\_\_\_\_

Recebe assistência estudantil?       Possui débitos na biblioteca?

**ENVIAR**

Figura 12 – Formulário de trancamento de matrícula

Fonte: Autor

#### 4.2.5 Deferimento de trancamento de matrícula

Sendo coordenador de curso desejo realizar o deferimento de uma requisição de trancamento de matrícula para finalizar a solicitação feita anteriormente;

Como acessar: estando autenticado, o usuário deverá navegar até a listagem de solicitações de trancamento de matrícula e expandir a solicitação a ser processada;

Campos: observações;

Resultado: Ao clicar no botão "Deferir" ou "Indeferir", os campos e botões relacionados ao deferimento serão desativados e um aviso será mostrado informando que a solicitação foi processada e um *e-mail* será redigido ao aluno solicitante, informando o mesmo do resultado.



**Solicitações de trancamento de matrícula** Filtrar ▾

Aluno	Matrícula	Curso	Data da solicitação
Isabel Carvalho	10070101	ADS	19/02/2021

**Detalhes**

CPF: 788.461.333-38  
 Ano/semestre de ingresso: 2020/1  
 Assistência estudantil: X  
 Débitos biblioteca: X  
 Licença maternidade: ▾  
 Alistamento militar: X

**Deferimento**

Observações

**DEFERIR**   **INDEFERIR**

Figura 13 – Seção de deferimento de solicitação de trancamento de matrícula

Fonte: Autor

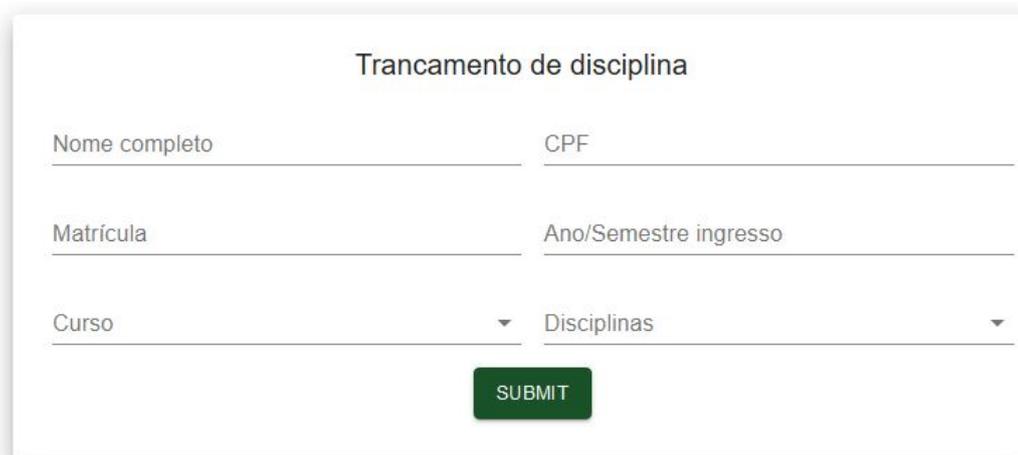
#### 4.2.6 Solicitação de trancamento de disciplinas

Sendo aluno desejo realizar a requisição do trancamento de uma ou mais disciplinas que estou matriculado no semestre corrente;

Como acessar: estando autenticado, o usuário deverá clicar no botão "Trancamento de disciplina";

Campos: Nome completo, número de matrícula, ano/semestre de ingresso, curso, CPF, disciplinas;

Resultado: Ao clicar no botão "Enviar", com os dados corretos, o formulário será reiniciado um aviso será mostrado informando que o requerimento foi enviado e um e-mail será redigido ao coordenador do curso especificado.



O formulário, intitulado "Trancamento de disciplina", contém os seguintes campos: "Nome completo" e "CPF" na primeira linha; "Matrícula" e "Ano/Semestre ingresso" na segunda linha; "Curso" e "Disciplinas" (com setas de seleção) na terceira linha. Abaixo dos campos, há um botão verde com o texto "SUBMIT".

Figura 14 – Formulário de trancamento de disciplinas

Fonte: Autor

#### 4.2.7 Deferimento de trancamento de disciplinas

Sendo coordenador de curso desejo realizar o deferimento de uma requisição de trancamento de disciplina para finalizar o requerimento feito posteriormente;

Como acessar: estando autenticado, o usuário deverá navegar até a listagem de solicitações de trancamento de disciplinas e expandir a solicitação a ser processada;

Campos: observações;

Resultado: Ao clicar no botão "Deferir" ou "Indeferir", os campos e botões relacionados ao deferimento serão desativados e um aviso será mostrado informando que a solicitação foi processada e um e-mail será redigido ao aluno solicitante, informando o mesmo do resultado.

Solicitações de trancamento de disciplina(s)			
Aluno	Matrícula	Curso	Data da solicitação
Bruno Lima	10070100	ADS	19/02/2021

**Detalhes**  
 CPF: 781.655.846-52  
 Ano/semestre de ingresso: 2019/2  
 Disciplinas: Fundamentos de computação

**Deferimento**  
 Observações

Figura 15 – Seção de deferimento de solicitação de trancamento de disciplinas  
 Fonte: Autor

#### 4.2.8 Solicitação de justificativa de faltas

Sendo aluno desejo realizar o envio de documentos que justificam presenças faltantes em uma disciplina que estou matriculado para que a falta não seja considerada no fim do semestre; Como acessar: estando autenticado, o usuário deverá clicar no botão "Justificativa de faltas"; Campos: Nome completo, número de matrícula, ano/semestre de ingresso, curso, data da falta, motivo, disciplinas, avaliação substitutiva; Resultado: Ao clicar no botão "Enviar", com os dados corretos, o formulário será reiniciado e um aviso será mostrado informando que o requerimento foi enviado e um e-mail será redigido ao coordenador do curso especificado.

### Justificativa de faltas

Nome completo	Matrícula
Ano/Semestre ingresso	Data da falta 01/01/2021
Curso	Motivo <span style="float: right;">▼</span>
	Disciplinas <span style="float: right;">▼</span>

Figura 16 – Solicitação de justificativa de faltas  
 Fonte: Autor

#### 4.2.9 Deferimento de justificativa de faltas

Sendo coordenador de curso desejo realizar o deferimento de uma requisição de justificativa de faltas para finalizar o requerimento feito posteriormente e, caso exista necessidade de

uma avaliação substitutiva, enviar uma notificação ao professor responsável pela disciplina;  
Como acessar: estando autenticado, o usuário deverá navegar até a listagem de solicitações de justificativa de faltas e expandir a solicitação a ser processada;

Campos: observações;

Resultados:

- Ao clicar no botão "Deferir" ou "Indeferir", os campos e botões relacionados ao deferimento serão desativados e um aviso será mostrado informando que a solicitação foi processada e um *e-mail* será redigido ao aluno solicitante, informando o mesmo do resultado;
- Ao clicar na opção "Notificar professor", um e-mail é redigido ao professor responsável pela disciplina, avisando o mesmo da necessidade de avaliação substitutiva do aluno solicitante.

Solicitações de justificativa de faltas				Filtrar
Aluno	Matrícula	Curso	Data da solicitação	
João Silva	10070100	ADS	19/02/2021	

<b>Detalhes</b> Data da falta: 15/08/2020 Motivo: Falecimento de parente Ano/semestre de ingresso: 2019/1	<b>Disciplinas:</b> <b>Fundamentos de computação</b> Avaliação substitutiva: X Professor: adm.temporario.ifrs@gmail.com <b>Redes de computadores I</b> Avaliação substitutiva: ✓ Professor: adm.temporario.ifrs@gmail.com <a href="#">NOTIFICAR PROFESSOR</a>	<b>Deferimento</b> Observações <input type="text"/> <input type="button" value="DEFERIR"/> <input type="button" value="INDEFERIR"/>
--	--	--

Figura 17 – Seção de deferimento de justificativa de faltas

Fonte: Autor

## 5 CONCLUSÃO E TRABALHOS FUTUROS

### 5.1 CONCLUSÃO

Com o objetivo de facilitar, agilizar e substituir as solicitações e requerimentos manuais e físicos feitos por alunos, assim como a análise dos mesmos, este trabalho desenvolveu uma solução baseada em microsserviços com fina granularização de funcionalidades, facilitando assim a implantação, manutenção, desenvolvimento de novos serviços e uso de diferentes tecnologias.

Foi possível concluir que uma arquitetura baseada em microsserviços possui um maior tempo para o entendimento completo do fluxo da aplicação como um todo, mas uma vez que essa dificuldade é ultrapassada, o tempo poupado com os pontos citados acima é extremamente alto, tornando o desenvolvimento significativamente mais simples e produtivo. Além disso, devido ao uso de tecnologias emergentes e amplamente utilizadas no mercado de trabalho, uma das possibilidades de contribuição deste trabalho seria o uso do mesmo em aulas de disciplinas de cursos relacionados à desenvolvimento de *software*, com o intuito de expandir as funcionalidades e posteriormente disponibilizá-las ao IFRS.

Com a inexistência de um sistema similar disponível, acredita-se que o uso do sistema proposto no dia-a-dia dos *campi* seria de grande ganho para a comunidade em geral.

### 5.2 TRABALHOS FUTUROS

Como trabalhos futuros foram identificadas as seguintes melhorias/funcionalidades:

- Implementação de todas as solicitações e requisições feitas por alunos não contempladas neste trabalho;
- Integração da aplicação com sistemas existentes do IFRS, sendo as principais delas:
  1. Autenticação de usuários;
  2. Validação de matrícula;
  3. Busca de professores;
  4. Busca de disciplinas;
  5. Busca de cursos.

Os itens listados acima (de 1 a 5) são considerados itens críticos e essenciais para uma implantação imediata do sistema (primeira versão de produção).

- Implantação da aplicação em infraestrutura própria do *campus* Restinga.

## REFERÊNCIAS

- AGARWAL, K. A study of virtualization overheads. **Stony Brook University**, p. 1–45, aug. 2015.
- DRAGONI, N.; GIALLORENZO, S.; LAFUENTE, A. L.; MAZZARA, M.; MONTESI, F.; MUSTAFIN, R.; SAFINA, L. Microservices: yesterday, today, and tomorrow. **HAL**, nov. 2017.
- FOWLER, M. **Microservices**. [s.n.], 2014. Disponível em: <<https://martinfowler.com/articles/microservices.html>>. Acesso em: 30 mar. 2020.
- JOSUTTIS, N. M. **SOA in Practice: The art of distributed system design**. Ca, United States Of America: O'reilly Media, Inc., 2007.
- LEWIS, J.; FOWLER, M. **What are microservices?** [s.n.], 2019. Disponível em: <<https://martinfowler.com/microservices/#what>>. Acesso em: 30 mar. 2020.
- MACHADO, M. **Micro Serviços: Qual a diferença para a Arquitetura Monolítica?** [s.n.], 2021. Disponível em: <<https://www.opus-software.com.br/micro-servicos-arquitectura-monolitica/>>. Acesso em: 10 abr. 2021.
- MACKENZIE, C. M.; LASKEY, K.; MCCABE, F.; BROWN, P. F.; METZ, R. Reference model for service oriented architecture. **OASIS**, ago. 2006.
- MEDEIROS, H. **Introdução ao MongoDB**. [s.n.], 2014. Disponível em: <<https://www.devmedia.com.br/introducao-ao-mongodb/30792>>. Acesso em: 30 mar. 2021.
- NADAREISHVILI, I.; MITRA, R.; AMUNDSEN, M.; MCLARTY, M. **Microservice Architecture: Aligning Principles, Practices, and Culture**. United States Of America: O'reilly Media, Inc., 2016.
- NEWMAN, S. **Building Microservices**. United States Of America: O'reilly Media, Inc., 2014.