

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA**  
**CAMPUS MONTEIRO**  
**CST EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**LUIZ MARIANO JÚNIOR**  
**BRUNO BEZERRA DOS SANTOS**

**TRABALHO DE CONCLUSÃO DE CURSO**  
**SPI: UM SOFTWARE PARA PESQUISA INSTITUCIONAL**

**MONTEIRO**  
**2023**

**LUIZ MARIANO JÚNIOR  
BRUNO BEZERRA DOS SANTOS**

**TRABALHO DE CONCLUSÃO DE CURSO  
SPI: UM SOFTWARE PARA PESQUISA INSTITUCIONAL**

**Trabalho de Conclusão de Curso (TCC)** apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Campus Monteiro, formatado na Modalidade Projeto de Implementação, como pré-requisito para a obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Me. Giuseppe Anthony Nascimento de Lima.

**MONTEIRO  
2023**

**LUIZ MARIANO JÚNIOR**  
**BRUNO BEZERRA DOS SANTOS**

**TRABALHO DE CONCLUSÃO DE CURSO**  
**SPI: UM SOFTWARE PARA PESQUISA INSTITUCIONAL**

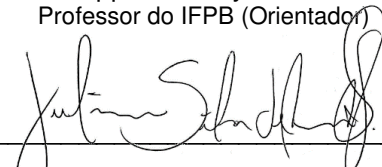
**Trabalho de Conclusão de Curso (TCC)** apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Campus Monteiro, formatado na Modalidade Projeto de Implementação, como pré-requisito para a obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Me. Giuseppe Anthony Nascimento de Lima.

**BANCA EXAMINADORA**



---

Prof. Me. Giuseppe Anthony Nascimento de Lima  
Professor do IFPB (Orientador)



---

Prof. Me. Julierme Silva de Araújo  
Professor do IFPB (Examinador)



---

Prof. Esp. Wagner de Oliveira Santos  
Professor do IFPB (Examinador)

Visto e permitida a impressão.

Monteiro-PB, 02 de outubro de 2023.



Documento assinado digitalmente  
WANDERLEY ALMEIDA DE MELO JUNIOR  
Data: 19/10/2023 17:53:57-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Me. Wanderley Almeida de Melo Júnior  
Coordenador do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Dados Internacionais de Catalogação na Publicação – CIP  
Bibliotecária responsável Porcina Formiga dos Santos Salgado CRB15/204  
IFPB Campus Monteiro.

M333t Mariano Junior, Luiz. ; Santos, Bruno Bezerra dos.

Trabalho de conclusão de curso SPI: um software para pesquisa Institucional / Luiz Mariano Junior; Bruno Bezerra dos Santos – Monteiro-PB. 2023.

48 fls. : il.

TCC ( Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB campus, Monteiro.

Orientador: Prof. Me. Giuseppe Anthony Nascimento de Lima.

1. Software - Desenvolvimento 2. Pesquisa Institucional – IFPB - Inconsistência 3. Rede Federal Educação Profissional, Científica Tecnológica – MEC I. Título.

CDU 004.453:004.451

## **AGRADECIMENTOS**

A elaboração deste trabalho não teria sido possível sem a colaboração e o apoio de diversas pessoas, a quem gostaríamos de agradecer:

Agradecemos aos nossos familiares, que sempre nos apoiaram e incentivaram ao longo dessa jornada.

Agradecemos aos nossos colegas de curso, especialmente a Igor Gusmão e a Ezequias Ferreira, que contribuíram conosco no desenvolvimento de partes do SPI, ao longo das disciplinas de Projeto I e II.

Agradecemos ao nosso orientador, o Prof. Giuseppe Lima, pela orientação e acompanhamento durante todo o processo de elaboração deste trabalho. Sua paciência, dedicação e conhecimento foram valiosas para logarmos êxito.

Agradecemos ao Procurador Institucional do IFPB, Francisco Fernandes de Araújo Neto, pela confiança e colaboração na obtenção dos dados e na validação dos requisitos necessários para o desenvolvimento da solução de software proposta neste trabalho, o SPI.

## RESUMO

Os dados acadêmicos desempenham um papel fundamental numa instituição de ensino, pois a partir deles é possível a geração e análise de indicadores e índices de eficiência institucional. Há que se garantir que os dados inseridos em um sistema de informação acadêmico possam ser periodicamente revisados, consistentemente, sobretudo quando geridos ou derivados de fontes com parâmetros diferentes. Este trabalho de conclusão de curso objetivou o desenvolvimento de um sistema *web* que fosse capaz de identificar inconsistências a partir dos dados do principal sistema de informação do IFPB (SUAP), que pertence à Rede Federal de Educação Profissional, Científica e Tecnológica (RFEPCT), considerando o estado de matrículas reportadas ao sistema oficial dessa rede, o SISTEC, que é controlado pelo Ministério da Educação (MEC). A aplicação *web* foi desenvolvida com recursos de facilitados de sumarização e visualização de eventuais inconsistências, aplicando-se soluções de extração, transformação e carregamento de dados (ETL), agregando-lhes confiabilidade e possibilitando utilizá-los para simular resultados de indicadores sob a plataforma Nilo Peçanha (PNP), em soluções futuras. Foi utilizada uma arquitetura de aplicação para processamento de grandes lotes de dados, oriundos do SUAP e do SISTEC, que foram comparados e validados, visando auxiliar o departamento de pesquisa institucional do IFPB, que poderá colaborativamente descentralizar esse esforço entre os seus *campi*. Relatórios simples de análise foram desenvolvidos, incluindo-se uma visualização gráfica para quantificá-los e classificá-los por estados de consistência.

**Palavras-chave:** Sistema Web; ETL; Pesquisa Institucional; SISTEC; Rede Federal de Educação Profissional, Científica e Tecnológica.

## ABSTRACT

Academic data play a fundamental role in an educational institution, because from them it is possible to generate and analyze institutional indicators and indices of performance. It must be ensured that the data inserted in an academic information system can be periodically revised, consistently, especially when they are managed or derived from sources with different parameters. This course conclusion work aimed the development of a web application that would be able to identify inconsistencies based on data from the main information system of the IFPB (SUAP), which belongs to the Federal Network of Professional, Scientific and Technological Education (RFEPCT), considering the status of enrollments reported to the official system of this network, known as SISTEC, which is controlled by Ministry of Education (MEC). The web application was developed with facilitated summary and visualization features over eventual inconsistencies, applying extraction, transformation and loading (ETL) data solutions to improve their reliability and to make possible the simulation indicators results under the Nilo Peçanha Platform (PNP), in future solutions. The application architecture was conceived to process large batches of data from SUAP and SISTEC, comparing and validating them to assist IFPB's institutional research department, which will be able to collaboratively decentralize this effort among its campuses. Simple analysis reports were developed, including a graphical view to quantify and classify them by their consistency statuses.

**Keywords:** Web Application; ETL; Institutional Research; Federal Network of Professional, Scientific and Technological Education.

## LISTA DE ABREVIATURAS E SIGLAS

<b>ADS</b>	Análise e Desenvolvimento de Sistemas
<b>CRUD</b>	<i>Create, Read, Update and Delete</i> (Criar, Ler, Atualizar e Remover)
<b>CSV</b>	<i>Comma-Separated Values</i> (formato de arquivo de exportação de dados, separado por vírgulas)
<b>DAO</b>	<i>Data Access Object</i>
<b>DTO</b>	<i>Data Transfer Object</i>
<b>ETL</b>	<i>Extract, Transform, Load</i> (Extração, Transformação e Carga de dados)
<b>IE</b>	Instituição de Ensino
<b>IFPE</b>	Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco
<b>IFPB</b>	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
<b>IFRN</b>	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
<b>MEC</b>	Ministério da Educação
<b>PI</b>	Pesquisador Educacional Institucional
<b>PNP</b>	Plataforma Nilo Peçanha.
<b>POC</b>	<i>Proof of Concept</i> (Prova de Conceito)
<b>RAM</b>	<i>Random Access Memory</i>
<b>REST</b>	<i>Representational State Transfer</i>
<b>RFEPCT</b>	Rede Federal de Educação Profissional, Científica e Tecnológica
<b>SUAP</b>	Sistema Unificado de Administração Pública
<b>UML</b>	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)



## LISTA DE QUADROS

Quadro 1 – Lean Canvas para o <i>software</i> SPI.....	15
Quadro 2 – Documentos inspecionados na análise de requisitos .....	22
Quadro 3 – Requisitos funcionais do SPI.....	23
Quadro 4 – Requisitos não-funcionais do software SPI .....	24
Quadro 5 – Iterações de desenvolvimento do SPI .....	36
Quadro 6 – Casos de testes sobre regras de negócio de “PesquisaService” .....	41
Quadro 7 – Distribuição dos testes de aceitação por telas do SPI.....	42

## LISTA DE FIGURAS

Figura 1 – Entidades Spring Batch.....	19
Figura 2 – Sequência de processamento de <i>chunks</i> no Spring Batch .....	20
Figura 3 – Diagrama de casos de uso.....	26
Figura 4 - Diagrama de classes conceitual .....	28
Figura 5 – Diagrama arquitetural.....	30
Figura 6 – Tela de <i>login</i> .....	31
Figura 7 – Tela de solicitação de acesso (por novos colaboradores institucionais) ..	31
Figura 8 – Tela de gerenciamento de cadastros de usuários.....	31
Figura 9 – Tela de listagem de pesquisas institucionais .....	32
Figura 10 – Tela de criação de uma pesquisa institucional.....	32
Figura 11 – Tela de sumarização de resultados de uma pesquisa institucional.....	33
Figura 12 – Tela de sumarização de resultados de uma pesquisa institucional (gráfico) .....	34
Figura 13 – Tela de relatórios de inconsistências .....	34
Figura 14 – Tela modal de seleção do tipo de relatório de inconsistência sobre uma pesquisa .....	34
Figura 15 - Tela de relatório de inconsistências do SUAP .....	35
Figura 16 - Tela de relatório de inconsistências do SISTEC .....	35
Figura 17 – Organização em pacotes do projeto <i>back-end</i> do SPI .....	38
Figura 18 – Organização do código front-end do SPI .....	39

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	11
1.1	Contexto e Problematização (Domínio do Problema)	11
1.2	Justificativa	12
1.3	Objetivos	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	15
2.1	Contextualização do Público-Alvo e das Aplicações do Produto de Software	15
2.2	Conceitos e Trabalhos Relacionados ao Domínio do Problema do Software	16
2.3	Extração, Transformação e Carregamento – ETL	17
2.4	Tecnologias Utilizadas	18
<b>3</b>	<b>RESULTADOS OBTIDOS</b>	22
3.1	Engenharia de Requisitos	22
3.2	Projeto Comportamental e Estrutural	24
3.3	Projeto Arquitetural	28
3.4	Projeto e Implementação da Interface com o Usuário	30
3.5	Projeto Gerencial do Software	36
3.6	Implementação do Protótipo	37
3.6.1	Back-end	37
3.6.2	Front-end	39
3.6.3	Projeto e Execução de Testes e Verificação de Qualidade	40
<b>4</b>	<b>CONSIDERAÇÕES FINAIS</b>	43
4.1	Revisão dos Objetivos	43
4.2	Trabalhos Futuros	44
	<b>REFERÊNCIAS</b>	45

## 1 INTRODUÇÃO

### 1.1. CONTEXTO E PROBLEMATIZAÇÃO (DOMÍNIO DO PROBLEMA)

Por meio do Ministério da Educação (MEC), o Governo Federal dispõe do Sistema Nacional de Informações da Educação Profissional e Tecnológica (SISTEC), que é uma plataforma de software para o registro e divulgação dos dados da Rede Federal de Educação Profissional, Científica e Tecnológica (RFEPCT), sob a qual também se validam os diplomas emitidos por seus cursos (MEC, 2018, p. 7).

Também pertencente a essa rede, o Instituto Federal da Paraíba (IFPB) utiliza o Sistema Unificado de Administração Pública (SUAP) que auxilia professores, técnicos administrativos e alunos na realização dos processos administrativos e acadêmicos (IFPB, 2022). Entre os seus diversos módulos, há um dedicado a fornecer aos funcionários do controle acadêmico e ao Procurador Educacional Institucional (PI) a possibilidade de geração de relatórios com informações dos estudantes e de sua situação nos cursos. Esses dados podem ser filtrados entre os *campi*, curso, modalidades e níveis de ensino, entre outros parâmetros.

Ciclicamente, o PI é o responsável por responder por essas informações, que são solicitadas pelo MEC, cujos desdobramentos impactam na composição da matriz orçamentária e no credenciamento para funcionamento de toda a instituição, assim como em avaliações externas, nas quais são atribuídos os conceitos de qualidade para toda a instituição ou os seus respectivos cursos. A partir do estado das matrículas é possível se mensurar a eficiência acadêmica geral da instituição, de um determinado campus ou de um curso. Portanto, o PI é o gestor encarregado por manter a coerência e a validade do cadastro de cursos, de seus ciclos de matrícula e das matrículas dos estudantes propriamente, nas respectivas bases de dados interna do IFPB (via SUAP) e do MEC (via SISTEC).

Ainda, a depender da abrangência das atividades de pesquisa institucional, o PI do IFPB pode precisar acionar a diretoria de controle acadêmico sistêmica, os coordenadores de controle acadêmico dos *campi* ou os coordenadores de cursos, visando o saneamento de inconsistências para a consolidação desses dados.

Contudo, o porte de uma instituição como o IFPB demanda o uso de ferramentas capazes de detectarem, com antecedência, quaisquer dessas ações saneadoras, de forma a apropriar os dados oriundos do SUAP perante os ciclos de

consolidação do SISTEC. Pode haver diferentes tipos de erros no cadastramento ou processamento de dados dessas fontes, a exemplo daqueles que não possuem uma correspondência direta de um sistema para o outro (estudantes, cursos, *campi*, dados situacionais de matrícula, entre outros) e até uma eventual duplicidade entre eles. Portanto, ocorre bastante esforço para detecção e correção desses dados, o que poderia ser realizado de forma assistida por software.

Desse modo, quaisquer cenários de consolidação desses dados a partir do SUAP e perante o SISTEC requerem uma verificação constante, até que se complete toda a janela de um ciclo de matrícula, determinada pelos seus prazos de envio. Uma solução por software possibilitaria antecipar as eventuais inconsistências e resolvê-las confiavelmente, em tempo hábil, independentemente da disponibilidade e de instabilidades na interface de carregamento provida pelo SISTEC, que eventualmente fica indisponível, sobretudo próximo do fim dos prazos de consolidação.

## 1.2 JUSTIFICATIVA

Perante o departamento de PI do IFPB, surgiu a demanda de interagir com ambos os sistemas SUAP e SISTEC, de forma que esse último forneça os indicadores de eficiência relacionados ao ensino, que são visualizáveis publicamente na plataforma Nilo Peçanha (MEC, 2023), a PNP.

A questão principal é que a PNP dispõe os dados de um ciclo anual muito tardiamente, do ponto de vista gerencial das instituições da RFEPCT. Não é raro os gestores indicarem que a possibilidade de antecipar isso por meio do tratamento dos dados de matrículas de seus sistemas acadêmicos (no caso do IFPB, oriundos do SUAP) proporcionaria a obtenção de recortes de temporalidade de granularidade mais fina (mensal, trimestral, semestral), viabilizando-se a obtenção de resultados antecipados para os setores competentes da IE, oportunizando-lhes o acionamento de procedimentos ainda dentro do efetivo fechamento do ciclo.

Considerando essas possibilidades, foi desenvolvido um projeto de software a partir da disciplina de Projeto I, do CST em Análise e Desenvolvimento de Sistemas, em que toda essa problemática foi investigada para a obtenção dos requisitos, do projeto interno, do protótipo da interface com o usuário e de provas de conceito de viabilidade tecnológica. Na sequência, na disciplina de Projeto II, o software foi

efetivamente implementado e testado como uma aplicação *web*, considerando o rol de requisitos negociadamente priorizados com o PI do IFPB para obtenção de uma primeira versão.

Portanto, este trabalho de conclusão de curso (TCC) se justifica como uma referência de documentação da solução obtida, em que foram revisadas as especificações do software desenvolvido, o seu modo de uso e a correção de falhas, abrangendo a aplicação de melhorias remanescentes da fase de desenvolvimento ocorrida nas disciplinas supramencionadas.

### 1.3 OBJETIVOS

#### 1.3.1 OBJETIVO GERAL

O objetivo deste trabalho foi a implementação do SPI (Sistema de Pesquisa Institucional), que é uma aplicação de software capaz de analisar dados exportados do SUAP e do SISTEC, aplicando-lhes um conjunto de regras para a identificação e quantificação de possíveis inconsistências, bem como de qual fonte de dados surgiram e possíveis causas.

#### 1.3.2 OBJETIVOS ESPECÍFICOS

Para alcance do objetivo geral, os seguintes objetivos específicos foram estabelecidos:

- Identificação das inconsistências entre os dados fornecidos, analisando os arquivos de exportação providos pelos sistemas SUAP e SISTEC, pela comparação de seus registros e do preestabelecimento de equivalências entre eles;
- Disponibilização de relatórios, para a visualização rápida de informações de inconsistências detectadas sobre os registros de matrículas, por ciclo SISTEC de consolidação;
- Aprovisionar o gerenciamento de pesquisas institucionais e do pessoal envolvido, a partir do cadastramento de pesquisas institucionais, com criação, leitura, atualização e deleção das mesmas, compondo-as a partir da importação dos arquivos providos pelo SUAP e SISTEC, considerando a colaboração de servidores credenciados do IFPB que representem os seus

respectivos *campi* na divisão do trabalho de alimentação e consolidação desses dados.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 CONTEXTUALIZAÇÃO DO PÚBLICO-ALVO E DAS APLICAÇÕES DO PRODUTO DE SOFTWARE

Para melhor compreensão do domínio de problema do SPI, nesta seção foi realizada uma análise para descrevê-lo de forma sintética, a partir do Lean Canvas, que é um modelo de análise de ideação de negócios criado por Maurya (2012). Os nove blocos do Lean Canvas para o SPI estão dispostos no Quadro 1.

**Quadro 1 – Lean Canvas para o *software* SPI**

<b>1. PROBLEMA</b>	<ul style="list-style-type: none"> <li>• Inviabilidade na detecção manual de inconsistências entre dados de cursos e de matrículas de estudantes já informados no SISTEC em relação ao SUAP, dado o volume de dados e de regras.</li> <li>• Indisponibilidade de monitoramento de indicadores sobre esses dados em ciclos menores, para o controle gerencial e orçamentário do IFPB, visto que somente ficam disponíveis depois do ano de referência na PNP.</li> <li>• Ausência de colaboração no processo de pesquisa institucional, dada a dificuldade de encaminhar e supervisionar a divisão do trabalho de pesquisa de inconsistências sob os estados de matrículas dos estudantes, entre os controles acadêmicos dos <i>campi</i> do IFPB.</li> </ul>
<b>2. SEGMENTOS DE CLIENTES</b>	<ul style="list-style-type: none"> <li>• Procuradores Institucionais (PIs).</li> <li>• Coordenadores de controles acadêmicos.</li> </ul>
<b>3. PROPOSTA ÚNICA DE VALOR</b>	<ul style="list-style-type: none"> <li>• Sistema <i>web</i> de fácil aprendizagem e uso com fluxo de processamento sob 03 bases de dados (SUAP, SISTEC ano anterior e ciclo) importadas pelo usuário, resultando na identificação de inconsistências de matrículas de um ciclo SISTEC para o outro.</li> </ul>
<b>4. SOLUÇÃO</b>	<ul style="list-style-type: none"> <li>• Mapeamento do estado de matrícula de estudantes entre a base SUAP, referenciando-a com a do SISTEC.</li> <li>• Relatórios de contabilização comparativos entre estados de matrículas oriundas do SUAP e do SISTEC, com as inconsistências encontradas.</li> </ul>
<b>5. CANAIS</b>	<ul style="list-style-type: none"> <li>• Divulgação do produto através dos fóruns de discussão da RFEPCT.</li> <li>• Página de documentação e de <i>download</i> no Github.</li> <li>• Grupo de colaboradores e de desenvolvedores para manutenção e aprimoramento do produto.</li> </ul>
<b>6. RECEITAS</b>	<ul style="list-style-type: none"> <li>• Editais de subvenção em pesquisa, desenvolvimento e inovação.</li> <li>• Licenciamento para outras IES.</li> </ul>
<b>7. ESTUTURA DE CUSTO</b>	<ul style="list-style-type: none"> <li>• Infraestrutura de nuvem.</li> <li>• Diárias e passagens na participação de fóruns e eventos da RFEPCT.</li> <li>• Desenvolvedores mantenedores do produto.</li> </ul>
<b>8. MÉTRICAS CHAVES</b>	<ul style="list-style-type: none"> <li>• Volume de <i>downloads</i>.</li> <li>• Volume de buscas na <i>web</i> sobre o produto.</li> <li>• Revisões positivas no Github.</li> <li>• Avaliações de usuários.</li> </ul>
<b>9. VANTAGEM COMPETITIVA</b>	<ul style="list-style-type: none"> <li>• Processo de <i>upload</i> de arquivos com configurabilidade pelo usuário no reconhecimento de campos de registros do sistema da IES, pelo SISTEC.</li> <li>• Possibilidade de fatiar as pesquisas por ciclos de matrícula SISTEC entre as unidades da instituição (<i>campi</i>), que ficam salvas e sobre controle de acesso restrito de pessoal credenciado.</li> </ul>

Fonte: Os autores.



O quadro modelo (*canvas*) é uma adaptação que vem do *Business Model Canvas* (Quadro de Modelo de Negócio) clássico e, segundo Maurya (2012), o principal objetivo é ajudar a descrever os principais conceitos dentro de uma perspectiva empreendedora, para uma constatação rápida e compreensiva do valor de um negócio em ideação.

## 2.2 CONCEITOS E TRABALHOS RELACIONADOS AO DOMÍNIO DO PROBLEMA DO SOFTWARE

O Pesquisador Institucional (PI) é a denominação atribuída através da Portaria nº 46, de 10 de janeiro de 2005 (MEC, 2005), que o define como um gestor responsável pelas atividades relacionadas com a informação de dados oficiais da instituição, relativas aos seus processos essenciais, como a geração de indicadores institucionais e a alocação de recursos da matriz orçamentária (UFRJ, 2023). Desde a Portaria Normativa nº 40, de 29 de dezembro de 2010 (MEC, 2010), o PI passou a ser chamado também de Procurador Educacional Institucional.

No IFPB, uma das principais ferramentas de trabalho do PI é o SUAP, que é um sistema originalmente desenvolvido pelo Instituto Federal do Rio Grande do Norte (IFRN), utilizado por diversos outros institutos federais. Existem várias funcionalidades atreladas ao SUAP, organizadas em módulos, transformando-o em um sistema completo e capaz de atender as particularidades de gerenciamento institucional e acadêmico (IFPB, 2016).

Considerando a apropriação de dados para o SISTEC a partir de sistemas acadêmicos, via identificação e correção dos mesmos, há disponível o aplicativo ARIA (IFPE, 2022), desenvolvido pelo Instituto Federal de Pernambuco (IFPE). Essa solução pode usar dados importados oriundos de outros sistemas acadêmicos recorrentes em instituições federais, destacando-se o SUAP, o Q-Acadêmico e o SIGAA (UFPB, 2022). Entretanto, é exigida a eventual compatibilização manual, modificando-se o rótulo da linha de cabeçalho dos arquivos exportados desses sistemas para que fiquem reconhecíveis pelo aplicativo. Outra constatação é a de que ele também não dispunha de nenhuma interface de mapeamento para realizar tal operação, exigindo que o PI realize a modificação das colunas todas as vezes, por conta própria, dos arquivos importados desses sistemas para análise de inconsistências.

Prospectando mais soluções similares, verificou-se ainda que não havia sido disponibilizada nenhuma ferramenta aberta com a capacidade de prever informações sobre o desempenho de instituições da RFEPCCT, a partir de indicadores de eficiência acadêmica da PNP, como retenção, evasão, entre outros.

### 2.3 EXTRAÇÃO, TRANSFORMAÇÃO E CARREGAMENTO – ETL

O desenvolvimento de soluções que combinam dados de diversas fontes, costuma exigir um processamento ao obtê-los e convertê-los para compatibilização, em que se cria um processo de transformação com regras bem definidas que se inicia antes de carregá-los em uma ou mais bases de dados consolidadas para que finalmente possam ser carregados posteriormente no uso em um sistema específico. Esse processo sobre os dados é conhecido como *Extract, Transform and Load* (ETL) (IBM, 2023a).

Essas etapas normalmente são aplicadas em um grande volume de dados. Contudo o procedimento costuma ser oneroso em termos de eficiência de software e hardware. Uma técnica comum é realizar essa sequência de processamento e armazenamento do conjunto de dados em lotes (batch), que divide os dados em blocos e faz com que o tempo de execução das tarefas diminua significativamente. Ainda, o processamento em batch possibilita uma melhor gestão de tolerância a falhas, pois caso o processamento de um lote específico tenha sido mal sucedido, há a possibilidade de identificá-lo e de se prosseguir com os demais lotes, oportunamente se reprocessando somente o bloco comprometido.

A primeira etapa de extração ETL (*extraction*) pode lidar com dados de diversas fontes (MICROSOFT, 2023a). Isso inclui bases de dados de esquemas diferentes, APIs, sites, planilhas e arquivos de formatos diversos, sendo uma das principais dificuldades integrá-los, sobretudo em grandes corporações. Cada situação exige um esforço diferente e várias são as soluções para facilitar o processo.

A segunda etapa, de transformação dos dados (*transform*), serve para mapeá-los e projetá-los antes que sejam salvos. Esse ponto é essencial, pois nele é quando acontece o processo de validação, autenticação, correção de registros duplicados e agregação dos dados (ORACLE, 2023). As regras de negócio definidas pelo executante são realizadas no sentido de sequenciar esse processo de

organização e combinação dos dados, que pode gerar novas entidades intermediárias para mapeamento, até se chegar nas entidades consolidadas (confiáveis e consultáveis).

Segundo a Amazon (2023), a terceira etapa ETL é a de carregamento (*load*), que pode acontecer de quatro formas distintas: (i) completa, quando todos os dados são carregados de uma única vez para o armazém de dados; (ii) incremental em intervalos regulares, registrando-se a data do último registro que foi efetuado para prosseguir para o próximo da sequência posteriormente; (iii) incremental por transmissão, indicado para um volume de dados pequeno com a finalidade de manter um fluxo de eventos controlados por meio de pipelines; e (iv) incremental em lotes, direcionado para um grande volume de dados, capaz de coletar alterações periodicamente, num período definido, em que nenhuma ação pode ocorrer entre os sistemas fonte e destino durante o processo.

## 2.4 TECNOLOGIAS UTILIZADAS

Para desenvolvimento do SPI como uma aplicação *web*, foram selecionadas as tecnologias mais utilizadas no momento, considerando dependências de software com um bom suporte da comunidade de desenvolvedores na *web* e aquelas em já se possuía alguma experiência, visando a diminuição de riscos.

Nesse sentido, o desenvolvimento da lógica da aplicação ocorreu sob a plataforma Java, a partir do *framework* Spring Boot Starter (SPRING, 2023a, 2023b), no qual se pode agregar convenientemente dependências base de uma aplicação no código do projeto, utilizando-se descritores de dependência prontos, conhecidos como inicializadores (*starters*). Por exemplo, adicionando-se apenas um *starter* específico, consegue-se obter uma aplicação Spring já com as dependências para mapeamento objeto relacional JPA (Java Persistence API), no lugar de se adicionar cada dependência necessária, uma a uma, sem o recurso.

O *starter* spring-boot-starter-web fornece suporte a aplicações *web*, incluindo o desenvolvimento de APIs RESTful (SPRING, 2023c). Nele há recursos para a gestão de requisições e respostas de serviços da lógica da aplicação, a partir do desenvolvimento de controladores que definem uma URI de identificação e acesso para cada serviço provido na lógica. Portanto, no esquema de rotas REST, combina-

se um dos métodos do protocolo HTTP mais um endereço que pode dispor de parâmetros de requisição.

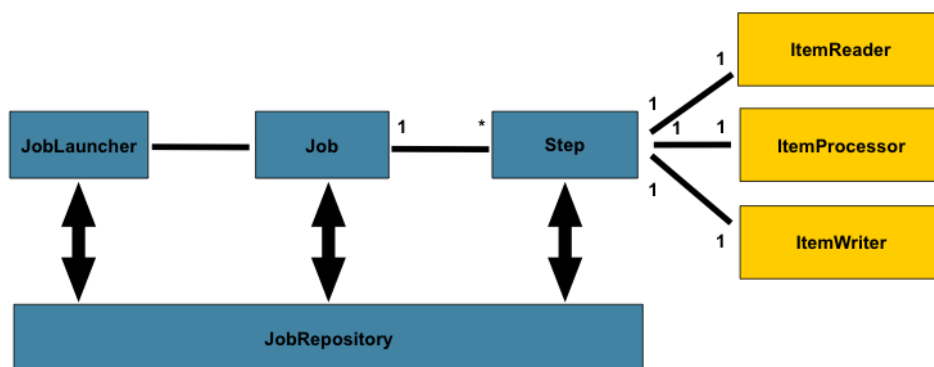
Já o *starter* `spring-boot-starter-security` fornece suporte à segurança em uma aplicação abrangendo autenticação e autorização, o que permite a proteção de recursos com mecanismos de segurança padrão, como autenticação básica ou autenticação por *token*, criptografia e proteção a ataques comuns (SPRING, 2023d).

O `spring-boot-starter-mail` foi utilizado fornecendo diversas configurações e interfaces que possibilitam a construção do corpo do e-mail como também a configuração simplificada para um servidor de e-mail externo (BAELDUNG, 2023).

Além desses *starters* mais comuns para o desenvolvimento *web* da lógica da aplicação, o SPI, por requerer processamento ETL, precisou adotar o `spring-boot-starter-batch` (SPRING, 2023e, 2023f). Previamente, haviam sido implementadas versões com as etapas de tratamento sem esse *starter*, obtendo-se um código com sérios problemas de coesão, considerando principalmente a modularização das etapas de transformação dos dados, tratando-se os registros das fontes de dados de uma só vez, sem ser em lotes.

Esse *starter* agrega escalabilidade e tolerância a falhas para processamento de registros, tratando-os com itens em lotes separáveis (*chunks*). Há um *framework* cuja arquitetura exige a implementação de objetos de execução de trabalhos de processamento configuráveis e gerenciáveis (objeto “Job”, vide Figura 1), que agregam objetos de etapa de execução (“Step”), cada uma contendo objetos com o código de extração de dados de uma fonte (“ItemReader”), objetos transformadores (“ItemProcessor”) e carregadores (“ItemWriter”), numa sequência bem definida.

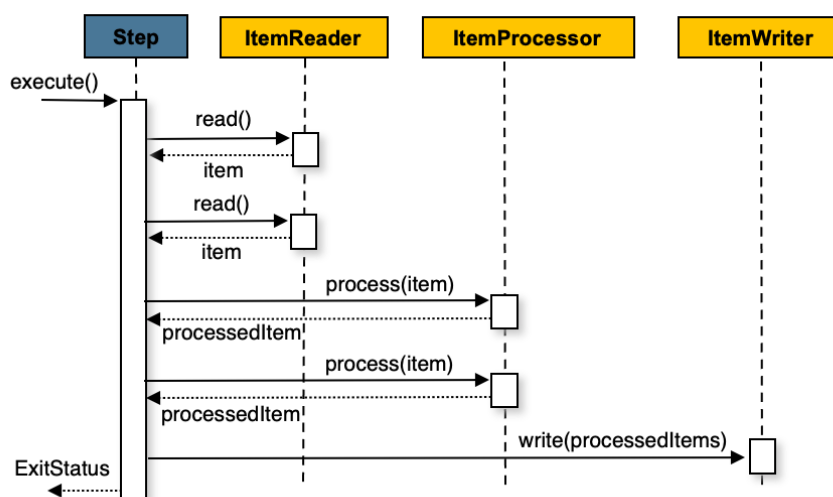
Figura 1 – Entidades Spring Batch



Fonte: SPRING (2023g).

Isso permite que se configure os *chunks* para que somente sejam efetivamente carregados ou persistidos se todos os itens de seu lote foram processados com êxito, por exemplo (Figura 2).

Figura 2 – Sequência de processamento de *chunks* no Spring Batch



Fonte: SPRING (2023h).

No armazenamento de dados, foi empregado o *starter* `spring-boot-starter-data-jpa` (SPRING, 2023i), que provê as dependências necessárias para persistência objeto-relacional por meio do padrão Java Persistence API, a partir do provedor Hibernate. O mapeamento de classes como objetos persistentes é facilitado por esse *starter*, que provê classes equivalentes ao padrão DAO, denominadas “Repository”, já com as operações de CRUD típicas implementadas. A conexão com o banco de dados se dá no arquivo de configuração da aplicação Spring, havendo também escrita mais fácil de consultas via objetos “Repository”, ocorrendo um para cada entidade persistente. Ainda, o PostgreSQL (2023) foi adotado como sistema de gerenciamento de banco de dados, por sua confiabilidade.

Combinadamente, para maior produtividade na validação de dados, adotou-se o `spring-boot-starter-validation`, que dispõe de dependências para identificação de dados de entrada providos por formulários antes de entregá-los para

processamento, evitando-se aqueles inválidos e inconsistentes pelas regras do negócio da aplicação.

Também foi utilizado o sistema de banco de dados orientado a documentos JSON, MongoDB (2023), que possui alta escalabilidade e velocidade. Isso foi útil no desenvolvimento do ETL do SPI, considerando planilhas utilizadas como fontes de dados, que possuem um alto volume de registros, cujas entidades geradas na transformação ETL se beneficiaram do recurso de persisti-las já de forma agregada (não exigindo junções em futuras consultas).

No contexto de *front-end* ou da camada de interface com o usuário do SPI, foi utilizado o ambiente de linguagem JavaScript, visto que é executável em qualquer navegador padrão. Mais precisamente, a interface com o usuário foi codificada na linguagem TypeScript (MICROSOFT, 2023b), a qual permite a tipagem de dados alternativamente ao JavaScript padrão, que não oferece esse tipo de suporte. No caso, um *script* implementado nessa linguagem é convertido compativelmente para JavaScript, oferecendo maior segurança de tipos para programadores.

O *framework* Next.js auxiliou a moldar o código do front-end. Ele provê melhor desempenho e acesso a uma variedade de recursos integrados, sem a necessidade de configuração de bibliotecas externas, removendo assim grande parte do esforço inicial necessário para o desenvolvimento de aplicações baseadas em React.js (VERCEL, 2023).

### 3 RESULTADOS OBTIDOS

#### 3.1 ENGENHARIA DE REQUISITOS

A fase de concepção do software SPI se iniciou com a execução de procedimentos de análise de requisitos, ao longo da graduação, na disciplina de Projeto I. Um dos principais objetivos nessa fase foi o de reunir as informações necessárias para obtenção dos requisitos funcionais e não-funcionais da aplicação *web*. Foram selecionadas técnicas de levantamento de requisitos, com a finalidade coletar informações sobre o domínio do problema e as necessidades e expectativas do cliente.

A primeira sessão de levantamento de requisitos foi uma entrevista roteirizada, com a duração de 1h30min, em que o cliente (PI do IFPB) descreveu o problema a ser solucionado, como também, propôs ideias de funcionalidades para o SPI.

Já a segunda sessão de levantamento focou na inspeção de alguns documentos importantes, utilizados na gestão do departamento de pesquisa institucional do IFPB. Ao longo de três semanas, a equipe identificou melhor as regras de negócio aplicáveis às funcionalidades do sistema, por meio da análise dos artefatos descritos no Quadro 2.

**Quadro 2 – Documentos inspecionados na análise de requisitos**

DOCUMENTO	DESCRIÇÃO
Manual do SISTEC para a RFEPCT	Orienta os servidores da rede a inserir informações de forma <i>online</i> na base de dados do sistema SISTEC
Guia de referência metodológica PNP 2020	Serve para auxiliar os gestores e pesquisadores em educação na utilização da PNP.
Tabela de regras de consistências PNP 2019	Lista e descreve quais são as regras de consistências e como são aplicadas dentro dos dados da PNP.
Planilha eletrônica de correspondência de status SUAP e SISTEC	Planilha eletrônica desenvolvida pelo PI do IFPB. Mapeia os estados de matrícula dos alunos, estabelecendo uma correspondência na rotulagem utilizada pelo SUAP, equiparando-a com a do SISTEC.

Fonte: Os autores.

Na terceira sessão de levantamento de requisitos aplicada foi uma oficina (*workshop*), com duração de duas horas. Nela, o cliente demonstrou como realizava as atividades para extrair das planilhas nas respectivas bases de dados do SUAP e

do SISTEC, manualmente, a verificação de possíveis inconsistências entre os registros dos alunos listados em ambas.

Ao longo da conclusão das sessões de levantamento de requisitos, ocorria a listagem das funcionalidades na matriz de requisitos funcionais (Quadro 3). A cada nova informação os já inclusos eventualmente eram renomeados, fundidos ou separados, considerando o critério de especificá-los: justificá-los com clareza e de expressar suas respectivas regras de negócio. Ainda, na elicitacão, houve a categorizaçã dos mesmos em agrupamentos de módulos e de submódulos, com a finalidade de facilitar a identificaçã de funcionalidades ainda ocultas. Estabilizada essa elicitacão, consolidou-se a especificaçã textual e validaçã de cada requisito.

**Quadro 3 – Requisitos funcionais do SPI**

MÓDULO	SUBMÓDULO	ID	REQUISITO FUNCIONAL
<b>CONTROLE DE ACESSO (AS)</b>	LOGIN (LG)	RF.AS.LG.01	Login e logout do sistema
	CADASTRO (CA)	RF.AS.CA.01	Solicitaçã de novo cadastro pelo colaborador institucional
		RF.AS.CA.02	Gerenciamento de colaborador institucional pelo PI
	RECUPERAÇã DE ACESSO (RA)	RF.AS.RA.01	Recuperaçã de senha
<b>ETL DE ANÁLISE DE DADOS (AD)</b>	GERENCIAMENTO DE PESQUISAS (GP)	RF.AD.GP.01	Gerenciamento de pesquisas institucionais por <i>campi</i>
	VERIFICAÇã DE INCONSISTÊNCIAS SUAP – SISTEC (VI)	RF.AD.VI.01	Comparar alunos
		RF.AD.VI.02	Filtrar ciclos de matricula
		RF.AD.VI.03	Comparar cursos
		RF.AD.VI.04	Comparar ciclos de matricula
<b>RELATÓRIOS (RL)</b>	RELATÓRIO SUMARIZADO (RG)	RF.RL.RG.01	Geraçã de relatório geral sumarizando a contabilizaçã de registros de matrícula consistentes ou não sobre as fontes de dados do SUAP e do SISTEC
	RELATÓRIO DETALHADO DE INCONSISTÊNCIAS (RI)	RF.RL.RI.01	Geraçã de relatório detalhado de ocorrências de matrículas inconsistentes, para registros SUAP e SISTEC.

Fonte: Os autores.



A elicitación de requisitos não-funcionais (Quadro 4) foi estabelecida com base na norma ISO/IEC 9126-1 (ISO/IEC, 2001). Destaca-se que ela já foi substituída pela norma ISO/IEC 25010 (ISO/IEC, 2011), mas a essa última obteve o acesso. Entretanto, ambas as normas definem um conjunto de áreas de qualidade interna e externa de um produto de software, auxiliando no processo elicitación de aspectos recorrentes, como usabilidade, eficiência, manutenibilidade, entre outros.

**Quadro 4 – Requisitos não-funcionais do software SPI**

CARACTERÍSTICA	SUBCARACTERÍSTICA	ID	REQUISITO NÃO-FUNCIONAL
FUNCIONALIDADE	SEGURANÇA	RNF 01	Privacidade nos dados dos usuários e nos dados que virão das planilhas SISTEC/SUAP.
	INTEROPERABILIDADE	RNF 02	Compatibilidade com arquivos CSV de importação de dados das fontes SUAP/SISTEC
	ERRO DO USUÁRIO	RNF 03	Validações feitas para impedir entrada de dados não esperadas
MANUTENIBILIDADE	TESTABILIDADE	RNF 04	Teste de aceitação simulando casos de uso na interface
	TESTABILIDADE	RNF 05	Teste unitário de funcionalidades
EFICIÊNCIA	COMPORTAMENTO EM RELAÇÃO AO TEMPO	RNF 06	Tempo de resposta na consulta dos dados importados
USABILIDADE	ATRATIVIDADE	RNF 07	Biblioteca de interface <i>web</i> com guia de estilo.
	APREENSIBILIDADE	RNF 08	Dicas de tela para funcionalidades do sistema
PORTABILIDADE	ADAPTABILIDADE	RNF 09	Acessível via navegadores padrão da <i>web</i>

Fonte: Os autores.

### 3.2 PROJETO COMPORTAMENTAL E ESTRUTURAL

Durante a análise de requisitos e o início da fase de projeto foram produzidos diagramas UML (Linguagem de Modelagem Unificada) com a finalidade de visualizar o domínio do problema melhor. Na fase de projeto propriamente, objetivou-se determinar cada uma das estruturas do software e as suas interdependências, considerando o desenvolvimento e implantação.

Na elicitación e análise de requisitos, foi realizado um refinamento dos requisitos funcionais derivados das sessões anteriormente descritas, a partir da

modelagem do diagrama de casos de uso do SPI. Entretanto, ressalta-se que ao longo do desenvolvimento esse diagrama foi sendo revisado e atualizado.

Um caso de uso é basicamente a descrição de um conjunto de ações que podem ser realizadas por meio de iterações entre atores (humanos ou sistemas automatizados) e o software em elaboração, com o objetivo de exemplificar para o cliente final quais seriam os requisitos funcionais do sistema (BOOCH; RUMBAUGH; JACOBSON, 2012). Um exemplo simples seria o de um sistema de caixa eletrônico de um banco, que ajuda a determinar os seus casos ou situações de uso, como “fazer saque”, “adicionar um depósito” e “fazer uma transferência”.

A modelagem de casos de uso UML, acompanhada da especificação dos requisitos funcionais servidos por cada caso, demonstrou ser suficiente para subsidiar o processo de prototipação e posterior modelagem interna do SPI. Dessa forma, os casos de uso não foram narrados por especificação textual e sim por meio da posterior prototipação da interface em si, iniciando a fase de projeto.

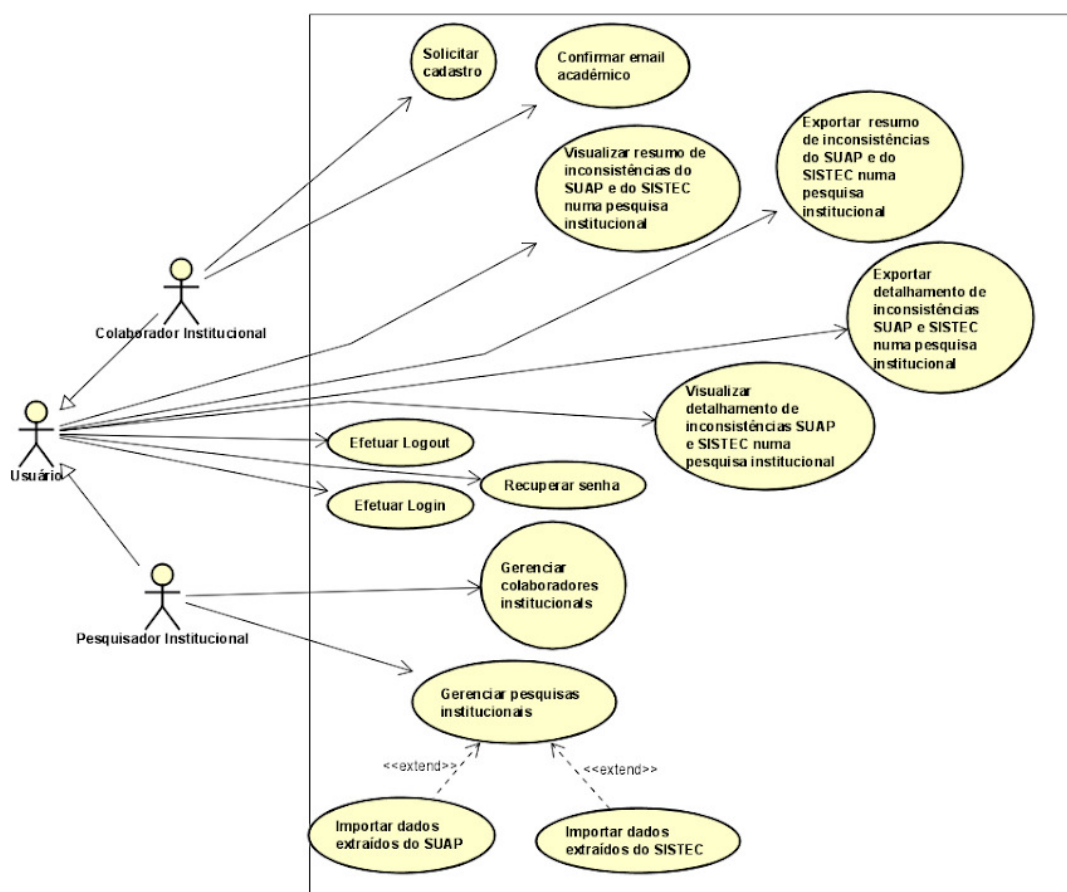
Conforme aponta Sommerville (2011), a prototipação pode ser definida como a construção de um modelo operacional do sistema para testar ideias e demonstrar funcionalidades aos usuários e clientes. Portanto, a prototipação de casos de uso permitiu a consolidação dos requisitos funcionais do sistema e de suas regras de negócio, identificando-se eventuais imprecisões e conflitos antes da implementação, a partir da previsão do quê, quando e como os seus atores usam o produto de software, numa linguagem clara e precisa, centrada nos usuários. Ainda, Pressman (2016) indica que a etapa de validação é o processo de avaliação do sistema ou componente para determinar se ele atende às especificações e requisitos definidos. Portanto, simular casos de uso garante um melhor alinhamento entre o que o software precisa oferecer e o que se espera dele.

A Figura 3 consolida a versão final com todos os casos de uso do SPI, havendo a presença de dois atores, o Pesquisador Institucional (PI) e o Colaborador Institucional.

Ambos atores “Pesquisador Institucional” (“PI”) e “Colaborador Institucional” podem realizar o caso de uso “Visualizar resumo de inconsistências do SUAP e do SISTEC numa pesquisa institucional”, que sumariza a quantificação de registros com e sem problemas de consistência, sobre cada uma das fontes de dados, SISTEC e SUAP. Também podem realizar o caso de uso de “Visualizar detalhamento de

inconsistências SUAP e SISTEC numa pesquisa institucional”, no qual os estudantes com inconsistências após o processamento ETL são listados distintamente entre as fontes de dados SISTEC e SUAP, verificando-se o rótulo do tipo de inconsistência de cada registro na perspectiva de sua respectiva fonte. Esses casos de uso possuem outros equivalentes, em que os atores podem exportar em formato de planilha eletrônica “CSV” o sumário de inconsistências ou as listagens com os registros inconsistentes SUAP e SISTEC, convenientemente. Entretanto, para o “Colaborador Institucional” os casos de uso de visualização e os de exportação somente abrangem registros concernentes ao seu campus de origem.

**Figura 3 – Diagrama de casos de uso**



Fonte: Os autores.

Destacam-se também, entre os casos de uso exclusivos do PI, o de “Importar dados extraídos do SISTEC” e de “Importar dados extraídos do SUAP”, que ocorrem quando ele cadastrar uma pesquisa institucional, a partir do caso de uso de CRUD

“Gerenciar pesquisas institucionais”. Nesse caso, o “PI” realiza a configuração das pesquisas, cada uma devendo corresponder a um determinado período de trabalho. Ainda, o “PI” pode realizar o caso “Gerenciar colaboradores institucionais”, que é o de cadastramento (CRUD) para gestão da concessão de acesso a representantes dos campi, em que pode habilitar ou não contas de colaboradores com e-mail confirmado.

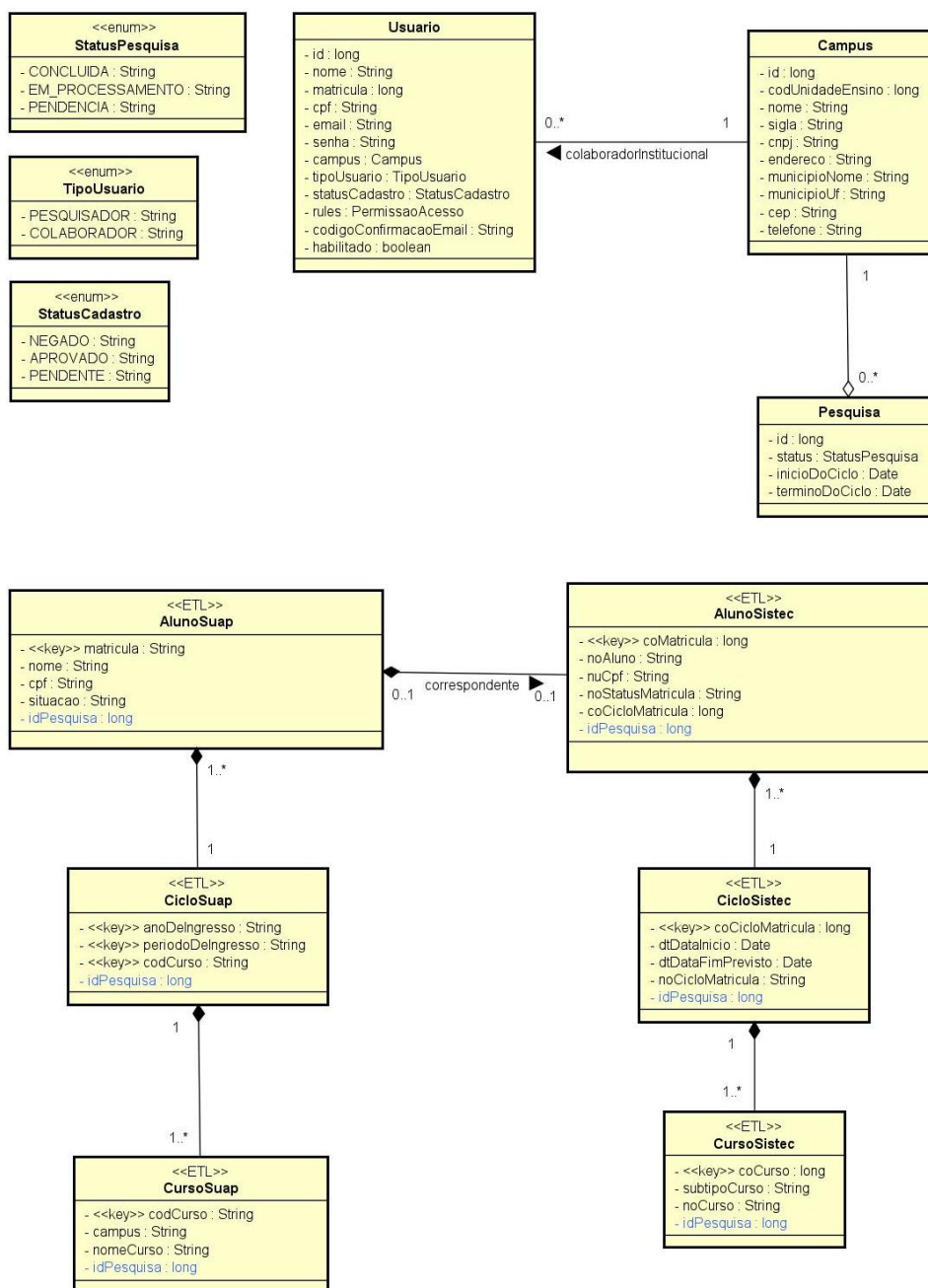
O “Colaborador Institucional” também pode optar por criar ele mesmo a sua conta, realizando o caso de uso “Solicitar cadastro”. Independentemente do método de criação desse tipo de conta, o “Colaborador Institucional” deve realizar o caso de “Confirma e-mail acadêmico” para que a conta se torne habilitada e o “PI” deve ter marcado a conta como ativa, no cadastro ou edição da conta em questão.

O domínio de problema foi modelado com um diagrama conceitual (Figura 4), de forma a identificar as principais entidades do SPI, tornando viável a determinação posterior do projeto estrutural da camada de lógica do negócio.

Ressalta-se que os *campi* da instituição já devem ser pré-cadastrados na instalação da aplicação, para que possam ser vinculados as contas de instâncias da classe “Usuario”, do tipo “COLABORADOR\_INSTITUCIONAL”, possibilitando que o SPI controle o acesso a visualizações e exportações do processo de ETL, direcionadamente. As classes marcadas com o estereótipo “<<ETL>>” correspondem as entidades intermediárias e resultantes do processo de transformação e carregamento ETL.

Consolidadamente, o SPI verifica apenas as coleções de entidades “AlunoSuap” e “AlunoSistec” no banco orientado a documentos, em que a primeira pode ou não agregar o aluno correspondente na fonte de dados da segunda, se ocorrer. Portanto, aqueles “AlunoSuap” sem agregação com um respectivo “AlunoSistec” são possíveis inconsistências e vice-versa. A estrutura dessas entidades de fato agrega todos os dados correspondentes ao ciclo de matrícula do curso de cada aluno (vide losangos preenchidos, nas associações todo parte, no lado da entidade todo ou aglutinadora, indicando composição UML). As entidades “<<ETL>>” possuem um campo identificador da pesquisa institucional a qual se aplicam (atributo “idPesquisa” em azul, na Figura 4), facilitando o referenciamento de buscas do rol de dados concernentes a uma pesquisa, que foram indexadas por este campo, para melhor desempenho.

Figura 4 - Diagrama de classes conceitual



Fonte: Os autores.

### 3.3 PROJETO ARQUITETURAL

A escolha da arquitetura correta para um projeto de software garante que o sistema seja eficiente, escalável e fácil de manter. Nesse sentido, o SPI foi concebido sob uma arquitetura *web* de três camadas: a de apresentação ou interface com o usuário; a de aplicação ou de lógica ou de serviços do negócio; e a de persistência de dados.

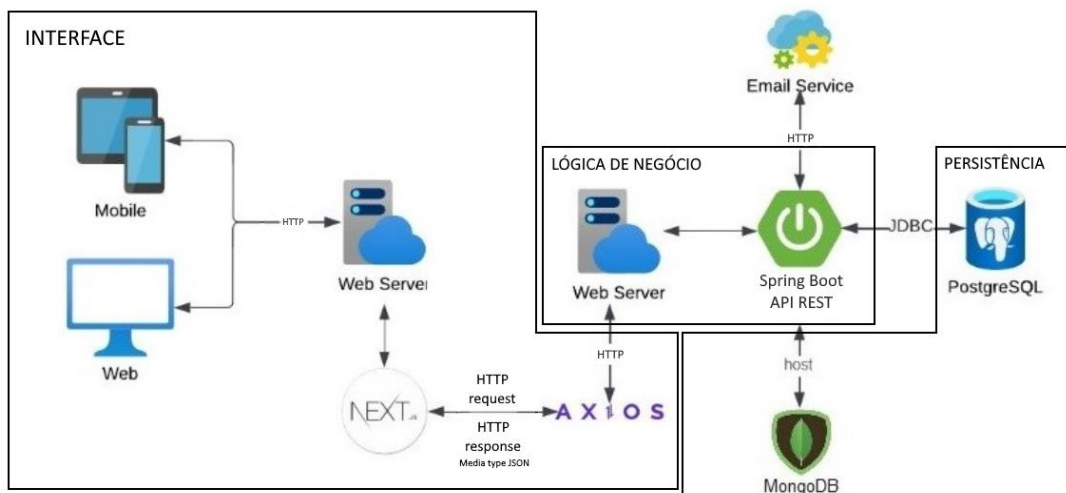
Segundo a IBM (2023b), os conceitos de camada (*layer*) e de nível (*tier*) devem ser compreendidos para esquematização de uma aplicação, respectivamente, equivalendo às divisões do software lógicas (divisão funcional) e às divisões físicas (componente da infraestrutura de execução). No SPI, como em arquiteturas *web* atuais, ocorrem as *layers* projetadas para serem executadas em *tiers* independentes, havendo: a *tier* do servidor *web*, contendo a *layer* da camada de apresentação; a *tier* do servidor de aplicação, onde é executada a lógica de negócios para processar entradas do usuário; e a *tier* do servidor do banco de dados, responsável por armazená-los. Dessa forma, pode-se realizar o desenvolvimento em equipes, paralela e separadamente, conferindo maior produtividade.

É habitual o entendimento de que a primeira camada compreende o *front-end* da aplicação e o de que as duas últimas compreendem o seu *back-end*. Mas, também, pode-se verificar que a camada da lógica de negócios é chamada por alguns como a de desenvolvimento *middle-end*. No SPI, a primeira e a segunda camadas foram moldadas, respectivamente, sob os preceitos arquiteturais disponibilizados pelos *frameworks* Spring Boot e Next.js (baseado em React.js).

A Figura 5 representa a visão dos componentes, conectores e subsistemas, descrevendo a arquitetura de implementação do SPI em alto nível. Nela há o *front-end*, (componente “INTERFACE”, na figura), que compreende todas as solicitações de ações requeridas pelo usuário através de um navegador típico (*browser*), que deve apresentar visualmente os dados de entrada e de saída. No *back-end*, verifica-se o alinhamento do componente “LÓGICA DE NEGÓCIO” com uma arquitetura orientada a serviços, a partir do fornecimento de uma API para o *front-end*, sob o protocolo REST, que permite a invocação de requisições de ações e transferência de dados sob o protocolo HTTP, não requerendo configurações extras sobre a infraestrutura de rede, como políticas de *firewalls*. Os serviços processam as solicitações originadas no cliente HTTP do *front-end* (AXIOS), ocorrendo o código com regras de validação sobre dados e transformações necessárias para persisti-los no servidor de banco de dados a partir da API REST do SPI (Spring Boot). Por fim, no componente “PERSISTÊNCIA”, de dados, foram aplicados dois bancos de dados, dada a necessidade de armazená-los através do processo de ETL a partir de

entidades agregadas (com o MongoDB) e de armazenar as entidades de gestão de pesquisas num esquema mais rígido ou relacional (com o PostgreSQL).

**Figura 5 – Diagrama arquitetural**



Fonte: Os autores.

### 3.4 PROJETO E IMPLEMENTAÇÃO DA INTERFACE COM O USUÁRIO

Para Buxton (2007), a utilização de protótipos de baixa-fidelidade, ou seja, protótipos que não possuem um alto nível de detalhamento gráfico ou possibilidade de simulação da interação, são essenciais e suficientes para validar as ideias sobre o desenvolvimento da interface com o usuário, sendo necessário desenhá-los, por meio de aplicativos de prototipação de UI ou mesmo papel e caneta.

O projeto da interface com o usuário do SPI foi realizado com o software de prototipação Balsamiq Wireframes (BALSAMIQ, 2023), focando-se no layout de componentes das mesmas e no sequenciamento da interação com as telas, para cada caso de uso. Essa prototipação foi iniciada na disciplina de Projeto I do curso, iterativamente junto ao cliente. Eventuais mudanças eram refletidas em requisitos e regras do negócio, a partir do *feedback* dos envolvidos. O protótipo final obteve 18 telas desenvolvidas.

Iniciando-se pela Figura 6 e Figura 7, que ilustram o aspecto das primeiras telas no processo de uso, serão demonstradas as demais telas já considerando a versão final da interface (codificada).

Figura 6 – Tela de *login*

Sistema de Pesquisa Institucional

E-mail:

Senha:

[Acessar](#)

[Esqueceu a senha?](#)

[Solicitar acesso](#)

Figura 7 – Tela de solicitação de acesso (por novos colaboradores institucionais)

Sistema de Pesquisa Institucional

SOLICITAR ACESSO

Nome:

Senha:

Matrícula:  CPF:

E-mail:  Campus:

[Solicitar](#) [Cancelar](#)

Fonte: Os autores.

A

Figura 8 é a tela do administrador (ator “PI”), o qual possui as ações de liberar (ou não) as solicitações de novas contas criadas por colaboradores institucionais.

Figura 8 – Tela de gerenciamento de cadastros de usuários

ADMINISTRADOR

Controle de Usuários

SOLICITAÇÕES PENDENTES [Voltar](#)

CPF	MATRÍCULA	NOME	AÇÕES
397.279.550-53	20182812291	Bruno Mariano	
570.805.560-07	201816029123	Bruno Santos	

Fonte: Os autores.

Após realizar o *login*, o usuário acessa a tela principal (Figura 9) que lista as pesquisas institucionais disponíveis, organizando-as em abas, de acordo com o estado de processamento ETL de cada pesquisa (“Progresso”, “Pendentes” e



“Completadas”). Nessa mesma tela também é possível: acompanhar o detalhamento de cada pesquisa, por meio da ação presente no botão com um ícone no formato de olho; excluir uma pesquisa, vide botão com um ícone de lixeira; e realizar o cadastro de uma nova pesquisa institucional (Figura 10).

**Figura 9 – Tela de listagem de pesquisas institucionais**

The screenshot displays the 'REGISTROS ACADÊMICOS' interface. On the left is a sidebar for the 'ADMINISTRADOR' with menu items: Pesquisas, Controle de Usuários, Relatórios, Sobre, and Sair. The main content area features a 'Nova Pesquisa' button and a table with the following structure:

Progresso (0)	Pendentes (1)	Completadas (0)						
<table border="1"> <thead> <tr> <th>CÓDIGO</th> <th>CICLO</th> <th>AÇÕES</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>2018.02 - 2022.12</td> <td>👁️ 🗑️</td> </tr> </tbody> </table>			CÓDIGO	CICLO	AÇÕES	3	2018.02 - 2022.12	👁️ 🗑️
CÓDIGO	CICLO	AÇÕES						
3	2018.02 - 2022.12	👁️ 🗑️						

Fonte: Os autores.

**Figura 10 – Tela de criação de uma pesquisa institucional**

The screenshot shows the 'NOVA PESQUISA' form. The sidebar on the left is identical to Figure 9. The main form area contains the following elements:

- Buttons: 'Cadastrar' (green) and 'Cancelar' (red).
- Section: 'NOVA PESQUISA' with subtitle 'PREENCHA OS DADOS PARA REGISTRAR A NOVA PESQUISA'.
- Fields: 'Início do ciclo:' and 'Fim do ciclo:' (date pickers).
- Field: 'Campus:' (dropdown menu).
- Buttons: 'IMPORTAR ARQUIVO' (dark grey) for 'Dados do ciclo:', 'Dados dos estudantes:', and 'Dados do suap:'.

Fonte: Os autores.

A Figura 11 ilustra a sumarização após o processamento ETL de uma pesquisa institucional, vide os *cards* retangulares “Alunos Duplicados”, “Alunos Divergentes” e “Em mais de uma Graduação”, que equivalem às inconsistências mais recorrentes. Na superior direita dessa mesma tela, há o botão de “Exportar”, que realiza a ação de gerar uma planilha com os dados das inconsistências encontradas e o botão de “Voltar” para a tela de listagem de pesquisas institucionais.

**Figura 11 – Tela de sumarização de resultados de uma pesquisa institucional**

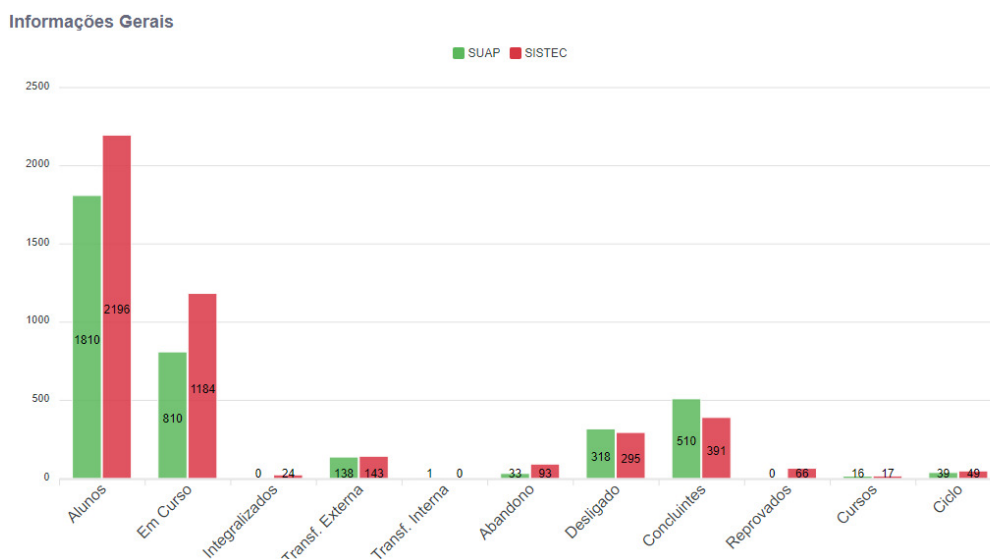


Fonte: Os autores.

Destaca-se que a tela acima é um recorte, em que, realizando-se a sua rolagem, pode-se observar todo o gráfico (vide Figura 12) que quantifica a pesquisa institucional, relacionando o quantitativo de matrículas processadas oriundas do SUAP e do SISTEC sob as rotulagens situacionais de matrículas de cada fonte.

Após o processamento de uma pesquisa institucional, se pode gerar o relatório de inconsistências no menu principal (vide Figura 13), havendo a seleção da respectiva pesquisa institucional processada, previamente criada (na Figura 10). Isso pode ser realizado pelo ícone de lista de tarefas sobre uma pesquisa processada, que é uma ação para ver as ocorrências de inconsistências, nos universos SUAP e do SISTEC, cuja escolha entre eles é realizada na tela modal, apresentada logo após o clique no ícone de lista de tarefas (Figura 14).

**Figura 12 – Tela de sumarização de resultados de uma pesquisa institucional (gráfico)**



Fonte: Os autores.

**Figura 13 – Tela de relatórios de inconsistências**

**ADMINISTRADOR**

- 🔍 Pesquisas
- 👤 Controle de Usuários
- 🟢 Relatórios
- ℹ Sobre
- 🔌 Sair

### RELATÓRIOS DE INCONSISTÊNCIA

[Voltar](#)

PESQUISA	CICLO	AÇÃO
3	2018.02 - 2022.12	

Fonte: Os autores.

**Figura 14 – Tela modal de seleção do tipo de relatório de inconsistência sobre uma pesquisa**



Fonte: Os autores.

Desse ponto, se chega às telas de relatórios SUAP (Figura 15) ou do SISTEC (Figura 16), que listam com paginação as entradas de matrículas, havendo a separação em abas de acordo com o tipo de inconsistência detectada e com a possibilidade de exportação em arquivo XLSX para maior conveniência do PI, se precisar rastrear algo com o aplicativo de planilhas eletrônicas de sua preferência, compatível com o Microsoft Excel.

**Figura 15 - Tela de relatório de inconsistências do SUAP**

CPF	NOME	MATRICULA	SITUAÇÃO	CÓDIGO DA MATRICULA
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	89902486
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	86047255
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	86054260
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	89902310
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	89902302
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	89902308
[REDACTED]	[REDACTED]	[REDACTED]	Concluída	89902288

Fonte: Os autores.

**Figura 16 - Tela de relatório de inconsistências do SISTEC**

CÓDIGO DO CICLO	CPF	NOME	MATRICULA	TOTAL
Nenhum Registro Encontrado.				

Fonte: Os autores.

### 3.5 PROJETO GERENCIAL DO SOFTWARE

Gerenciar um projeto de software exige organização, planejamento, colaboração e liderança. Metodologicamente, o SPI foi gerenciado ao longo de três fases, com focos diferentes na evolução do produto.

A primeira fase do projeto ocorreu durante a disciplina de Projeto I, cujo processo de software adotado assemelhava-se a um processo evolucionário, visando a concepção do software por iterações de modelagem ou de prototipação. Ao longo de 16 semanas, eram revisados semanalmente os riscos visando o refinamento do produto, até se chegar em uma prova de conceito (POC, do inglês “*proof of concept*”). A fase POC (últimas quatro semanas) auxiliou na consolidação da arquitetura e no aprendizado das tecnologias designadas, sobretudo as de processamento ETL, que culminaram na decisão de utilização do Spring Batch.

Na segunda fase, durante a disciplina de Projeto II, o processo se tornou incremental, havendo a entrega em blocos de partes funcionais, ao longo de 06 iterações (*sprints*) de 15 dias (vide Quadro 5). Esse processo foi baseado em atividades do Scrum *framework* (SCHWABER; SHUTERLAND, 2020), já que a equipe não empregou as atividades de *spring retrospective* ao final das *sprints* e o *daily scrum* com a frequência recomendada, pois as interações entre os membros ocorreram de forma assíncrona, por meio de *chat*, majoritariamente. O desenvolvimento nessa fase prosseguiu centrado no usuário, em estreita colaboração com o cliente, em atenção as suas necessidades e expectativas.

**Quadro 5 – Iterações de desenvolvimento do SPI**

ITERAÇÃO	OBJETIVOS	REQUISITOS
1	<ul style="list-style-type: none"> <li>• Possibilitar cadastro de novos usuários (solicitação de acesso)</li> <li>• Fornecer <i>login</i> e <i>logout</i> do sistema</li> <li>• Disponibilizar recuperação de senha</li> </ul>	RF.AS.LG.01, RF.AS.CA.01 e RF.AS.RA.01
2	<ul style="list-style-type: none"> <li>• Fornecer o controle do cadastramento de novos usuários por meio das solicitações de acesso</li> </ul>	RF.AS.CA.02
3	<ul style="list-style-type: none"> <li>• Fornecer pesquisas institucionais (<i>back-end</i>)</li> <li>• Validar fonte de dados importadas quanto a sua consistência e normalização.</li> </ul>	RF.AD.VI.01, RF.AD.VI.02, RF.AD.VI.03, RF.AD.VI.04
4	<ul style="list-style-type: none"> <li>• Disponibilizar relatórios de inconsistências e <i>dashboard</i> (sumário de pesquisas)</li> </ul>	RF.RL.RG.01, RF.RL.RI.01
5	<ul style="list-style-type: none"> <li>• Ajustes em pesquisas institucionais</li> </ul>	RF.AD.GP.01
6	<ul style="list-style-type: none"> <li>• Ajustes em pesquisas institucionais</li> </ul>	RF.AD.GP.01

Fonte: Os autores.

A terceira fase, que durou um mês e meio, focou em mais refinamentos de itens na interface com o usuário, na revisão das especificações e do código, com aprimoramentos baseados em indicações do cliente e representantes, sobretudo relacionados com os submódulos de requisitos RI e RG.

### 3.6 IMPLEMENTAÇÃO DO PROTÓTIPO

Nesse tópico será descrito como foram organizados os módulos e componentes do SPI em nível de código de *back-end* e de *front-end*, visando desenvolvimento contínuo com código legível, fácil de manter e reutilizável.

A gerência de configuração do SPI foi assistida pela plataforma GitHub (2023), havendo 02 repositórios, um para o *back-end*<sup>1</sup> e outro para o *front-end*<sup>2</sup>. Ela permite a aplicação de versionamentos sobre o código, o controle de mudanças e o gerenciamento de problemas decorrentes de integrações e de revisões do código entre o time de desenvolvedores.

#### 3.6.1 BACK-END

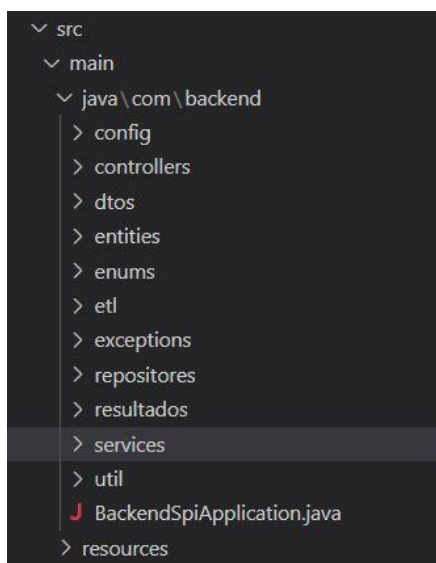
Podemos observar na Figura 17 como o código do *back-end* foi organizado, seguindo um conjunto de definições recorrentes em aplicações do *framework* Spring Boot, refletidas na divisão em pacotes de classes Java, destacando-se: o pacote “entities”, com as classes do negócio; o “controllers”, com os *endpoints* de serviço REST; o “services”, com classes do padrão de projeto orientado a objetos *Façade*, para acesso a módulos de serviços da lógica do negócio; “etl”, com as classes necessárias para processamento ETL com Spring Batch; e “repositores”, pacote responsável pela camada de persistência do Spring Data.

Considerando a transferência de dados na API REST implementada nos *endpoints* presentes em classes do pacote “controllers”. Classes de objetos sob o padrão DTO (*Data Transfer Object*) foram criadas no pacote “dtos”, visando uma serialização mais restrita do rol de atributos sensíveis de entidades em JSON destinados à apresentação na camada de interface com o usuário.

---

<sup>1</sup> Disponível em: <<https://github.com/Brunobzrra/SPI/tree/main/back-end>>

<sup>2</sup> Disponível em: <<https://github.com/Brunobzrra/SPI/tree/main/front-end>>.

**Figura 17 – Organização em pacotes do projeto *back-end* do SPI**

Fonte: Os autores.

O maior esforço na implementação da lógica da aplicação ocorreu no processamento ETL, que inicialmente foi experimentado sem assistência de nenhum *framework*, obtendo-se um código com baixíssima coesão quanto à distribuição do código de cada uma das fases de processamento, pois ficou extenso e difícil de manter. Esse componente foi todo reescrito com as interfaces de tipos do *framework* Spring Batch. Entre as duas versões, foi verificado também o tempo de execução do processamento ETL, cujos testes foram realizados em um computador pessoal *desktop*, com 8 GB de memória RAM (*Random Access Memory*) e processador Intel Core i5, com a frequência de operação base de 2,3 GHz.

A implementação sem Spring Batch do ETL atingiu 07 segundos, abrangendo de 13 mil registros de entrada em formato CSV. O tempo de processamento atingido foi o menor, verificado pela inexistência cargas intermediárias na transformação dos dados, que eram limpos e pós-processados em memória e somente carregados em arquivos ao final do todo o processo de ETL. Foi aplicada uma técnica de escrita simples no formato JSON, dentro de um diretório com arquivos específicos para cada tipo de relatório previsto para a aplicação.

Com a implementação do ETL com Spring Batch, o tempo de execução do software aumentou significativamente para 47 segundos, devido a necessidade de armazenamento em coleções do MongoDB, desde a primeira etapa de

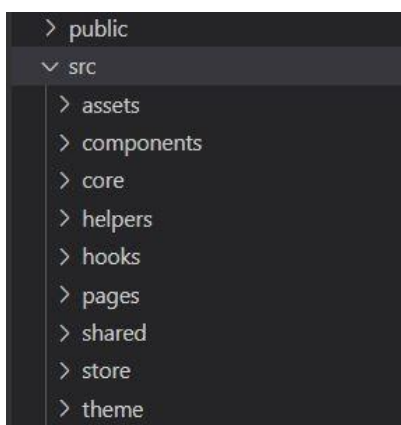
transformação. Esse processamento foi configurado em lotes de 100 registros. Ainda as etapas que exigiam cruzamento entre registros SISTEC e SUAP exigiam consultas sobre registros já transformados e carregados e a escrita daqueles registros ainda não carregados.

Apesar do tempo maior com o Spring Batch, ressalta-se a melhor percepção de distribuição das etapas ETL no código, que contribuiu para a sua maior inteligibilidade e facilitação dos testes de sequências de processamento. Ainda, a segunda versão demonstrou ser melhor, dada a possibilidade de se gerir e de se dimensionar os lotes de processamento, com possibilidade posterior de reprocessá-los em caso de falhas.

### 3.6.2 FRONT-END

A Figura 18 apresenta a organização do código *front-end* desenvolvido com o *framework* Next.js. A estrutura de pastas seguiu as convenções de uma aplicação React.js típica, visando uma melhor distribuição e manutenção dos *scripts* da aplicação.

**Figura 18 – Organização do código front-end do SPI**



Fonte: Os autores.

Em destaque, há o diretório “pages”, que contém 12 *scripts* de páginas, cada um responsável por aglutinar ou combinar os componentes de interface necessários em telas, que são endereçáveis como rotas da aplicação. Seguiu-se a boa prática de organizar as páginas em subpastas com nomes que correspondem a essas rotas, que são pontos de acesso acessíveis por uma URL específica, capaz de direcionar a



uma dada tela na navegação. Também foi disposta a pasta “components”, contendo 11 componentes dispostos em subpastas, cada uma correspondendo a uma módulo com finalidades comuns, permitindo o reuso de componentes de apresentação entre as páginas da aplicação. Por exemplo, há a subpasta “layout”, contendo componentes que dividem a tela, como o “Menu”, “Card”, “Modal”, entre outras.

Ainda, o diretório “assets” contém *scripts* responsáveis por armazenar os arquivos estáticos, como imagens, fontes e animações. Já a pasta “shared” contém a subpasta “service”, cujo nome, por convenção, descreve um diretório com *scripts* que realizam chamadas à API REST do *back-end* do SPI. Esses *scripts* são chamados por páginas (e eventualmente componentes específicos) que abrangem leitura ou consolidação de dados sob os serviços providos pela lógica do negócio.

### 3.6.3 PROJETO E EXECUÇÃO DE TESTES E VERIFICAÇÃO DE QUALIDADE

A realização de testes possibilita a identificação de erros durante as etapas de desenvolvimento e garante melhor as funcionalidades para o cliente (SOARES, 2020). Os testes servem para garantir a qualidade do sistema e assegurar o seu funcionamento e conformidade. No SPI, foram aplicados testes unitários, de integração e de aceitação.

Para testes unitários, foi utilizado o *starter* spring-boot-starter-test (SPRING, 2023j), o qual reúne as dependências para a execução de testes automatizados, incluindo a biblioteca JUnit, mantida pela JUnit Team (2023). Com ela os testes unitários podem ser facilmente organizados em suítes de testes, com bom suporte por documentação. Ainda nesse sentido, foram desenvolvidos alguns testes de integração para a verificação de incompatibilidades entre as funções internas que foram desenvolvidas isoladamente, mas que, na hora da execução, devem se combinar sem conflitos.

O Quadro 6 ilustra parte da especificação dos testes, entre os que foram realizados na camada lógica da aplicação, com alvo principal sobre a classe de serviço “PesquisaService”, já que ela é a responsável pelo CRUD de pesquisas institucionais e por também executar as regras do negócio de ETL do SPI.

**Quadro 6 – Casos de testes sobre regras de negócio de “PesquisaService”**

<b>REGRA</b>	<b>TESTE</b>
Cadastrar uma pesquisa passando alguma planilha nula.	Ao cadastrar pesquisa enviar três arquivos de planilhas, caso alguma não seja enviada, emitir a exceção “ARQUIVO (NOME DA PLANILHA) NÃO FORNECIDO”.
Importar planilhas de pesquisa com um arquivo de planilha de formato diferente do CSV.	Não havendo correspondência com o tipo CSV, retornar a exceção “ARQUIVO (NOME DA PLANILHA) NÃO É NO FORMATO CSV”.
Cadastrar uma pesquisa passando datas inválidas para o ciclo.	Por definição, a data de início do ciclo não deverá ser superior à data de término, como também não será permitido data iguais. Se algum dos casos citados acontecerem, o sistema deverá emitir uma exceção no formato “DATA DE TÉRMINO INVÁLIDA EM RELAÇÃO A DATA DE INÍCIO”.
Cadastrar uma pesquisa com a planilha faltando campo necessário para realizar a extração ETL.	Os arquivos importados na pesquisa devem conter, no caso de planilhas de alunos do SISTEC, as colunas: “NO_ALUNO”, “NU_CPF”, “CO_MATRICULA”, “CO_CICLO_MATRICULA”, “DT_DATA_INICIO”, “DT_DATA_FIM_PREVISTO” e “NO_STATUS_MATRICULA”.  Para a importação da planilha contendo os alunos do SUAP, detectar os campos: “Matricula”, “Nome”, “Curso”, “Situacao”, “Ano de ingresso”, “CPF” e “Período de ingresso”.  Para a importação da planilha contendo os ciclos de matrículas do SISTEC, devem ocorrer os campos: “SUBTIPO CURSOS”, “CÓDIGO CURSO”, “NOME DO CURSO”, “CÓDIGO CICLO DE MATRÍCULA”, “CICLO DE MATRÍCULA”, “DATA INICIO DO CURSO” e “DATA FIM PREVISTO DO CURSO”.  Não importa a ordem em que os campos ocorram nas planilha importadas. Havendo incorrespondências, emitir a exceção “COLUNA (NOME DA COLUNA) DO ARQUIVO (NOME DO ARQUIVO) NÃO FOI ENCONTRADA”.
Cadastrar pesquisa com algum valor vazio nas colunas necessárias para o mapeamento.	Esse teste é caracterizado pela detecção de valores, que são obrigatórios sobre os campos das planilhas importadas. Emitir a exceção “ERRO NA LINHA (NÚMERO DA LINHA) DO ARQUIVO (NOME DO ARQUIVO): CONTÉM VALOR VAZIO EM SUA TUPLA”.
Cadastrar pesquisa, combinando-se as regras anteriores	Teste correspondendo a verificação de todas as regras na criação de uma pesquisa institucional e no processamento de seus arquivos SISTEC e SUAP, verificando retorno se êxito se todas as condições foram respeitadas.

Fonte: Os autores.

Considerando os testes de aceitação do sistema, eles foram realizados por meio da simulação do uso pelos desenvolvedores, em conformidade com os representantes do cliente, ao final de cada iteração de desenvolvimento. Eventuais ajustes e aprimoramentos foram aplicados para adequação. O Quadro 7 lista as o conjunto de telas da interface, que roteirizaram o percurso de realização desses testes.

**Quadro 7 – Distribuição dos testes de aceitação por telas do SPI**

<b>TELA</b>	<b>DESCRIÇÃO</b>
Tela de login	Essa tela é responsável pelo <i>login</i> (entrada) do usuário no sistema
Tela de inicial do sistema	Ao entrar no sistema essa é primeira tela que o usuário visualizará no sistema, que dispõe a listagem de pesquisas visíveis para o usuário.
Tela de cadastrar uma nova pesquisa	Essa tela é responsável por criar uma nova pesquisa. O usuário poderá fazer o <i>upload</i> das planilhas dos arquivos SISTEC e SUAP, inserir as datas de início e término dos ciclos que serão utilizadas como parâmetro de processamento das planilhas e informar qual o campus de origem dos dados.
Tela de exibir os resultados de uma pesquisa	No final da criação de uma nova pesquisa, após o processo de ETL ser concluído, automaticamente irá aparecer uma tela com os resultados gráficos que representam os valores obtidos, quanto às principais métricas solicitadas pelo cliente da respectiva pesquisa.
Tela de exibição dos relatórios SISTEC e SUAP	O acesso a essa tela fica no menu esquerdo da tela inicial do sistema. Após a criação de uma pesquisa, os seus relatórios ficam listados em tela. O usuário poderá acessar os relatórios e exportá-los em formato XLSX.
Tela sobre o sistema	Essa tela descreve as informações do sistema, os seus desenvolvedores e parceiros.
Tela de solicitação de acesso	A tela solicitação de acesso fica disponível para usuários que fazem parte dos controles acadêmicos de cada campus. Para que seja efetuada a solicitação de acesso ao sistema, é necessário que o solicitante tenha um e-mail acadêmico válido e que seja liberado pelo PI.
Tela de liberação de acessos	Essa tela possibilita ao PI, aceitar ou recusar solicitações efetuadas através da listagem das requisições dos usuários que queiram ter acesso ao sistema.
Tela de recuperação de senha	Essa tela serve para todos os usuários que tem acesso ao sistema, mas que por algum motivo esqueceram sua senha.

Fonte: Os autores.

Outro aspecto a ser enfatizado são os testes de segurança dos *endpoints* do SPI, especialmente por ser uma aplicação que lida com dados sensíveis. O controle de acesso aos *endpoints* é uma das principais medidas para garantir a segurança da aplicação. Na autenticação o tipo de usuário é identificado, havendo a autorização de acesso aos recursos específicos em nível de cada um dos *endpoints* da API REST do SPI. Cada *endpoint* sensível por controle de acesso foi verificado, simulando-se a sequência de requisições possíveis ou não, sobre uma sessão de autenticação para o usuário “PI” e para um “Colaborador Institucional”.

## 4 CONSIDERAÇÕES FINAIS

A aplicação *web* SPI é uma solução de software idealizada para promover agilidade no processamento e dados acadêmicos de matrículas de uma instituição da RFEPCT, utilizando o IFPB como estudo de caso. As funcionalidades obtidas permitiriam validar esses dados quanto a sua consistência, incluindo-se verificações de duplicidade, sob a perspectiva das regras institucionais do IFPB e da regulamentação federal vigente, quando inclusos ou apropriados no SISTEC.

Portanto, foi desenvolvido e experimentado um processamento ETL viável sobre esses dados, utilizando-se o Spring Batch como tecnologia base, a partir da (i) extração de registros e ciclos de matrículas provenientes de arquivos CSV (importados das fontes SUAP e SISTEC), (ii) da transformação desses dados, com regras de limpeza, correção e de correspondências entre eles; e (iii) da carga desses registros, considerando cada aluno, no SUAP e no SISTEC, eventuais combinações entre eles e a indicação de inconsistências entre os mesmos em ambas as bases de dados.

Entretanto, era previsto o provimento de recursos relacionados com a simulação dos indicadores principais da Plataforma Nilo Peçanha (PNP), que não foram desenvolvidos devido ao cronograma do projeto e priorização pelo PI do IFPB, que optou obtê-la a partir de outras ferramentas de *business intelligence* (destacando-se o Metabase), que poderiam ser alimentadas com os dados já exportados pelo SPI.

### 4.1 REVISÃO DOS OBJETIVOS

O software foi disponibilizado como um mínimo produto viável para o PI do IFPB, com o qual foi possível verificar a capacidade de identificação automatizada de inconsistências em matrículas do SUAP para o SISTEC e vice-versa.

Também, como resultado da análise do processo executado anteriormente, foi possível realizar um demonstrativo do processamento das informações e da identificação de possíveis ocorrências quanto ao volume de dados tratados e de matrículas inconsistentes, por meio da disponibilização de relatórios na visão de cada fonte de origem de dados, SUAP e SISTEC.

Por fim, foi possível atingir o objetivo de gestão de pesquisas institucionais de forma colaborativa, com atribuição de cadastros de acessos a serem realizados pelos próprios colaboradores institucionais nos *campi*, com liberação pelo PI, bem como a criação de pesquisas institucionais pelo mesmo em blocos, por campus.

#### 4.2 TRABALHOS FUTUROS

Para disponibilização do SPI em ambiente de produção, ressalta-se a necessidade de desenvolvimento de mais cenários de teste, especialmente quanto à verificação de cargas de dados maiores na importação SISTEC e SUAP, também de forma simultânea, por vários usuários. Mesmo considerando a solidez do Spring Batch, seria interessante averiguar, nessas situações, as capacidades e o comportamento final de carga.

Também, considerando as especificações de código, ainda há a possibilidade de documentação dos *endpoints* REST da API, a partir da ferramenta Swagger, que permite a verificação e simulação de requisições.

Indica-se também uma fase de homologação da aplicação antes da sua disponibilização final, considerando o interesse do departamento de PI do IFPB para que colaboradores institucionais também possam receber treinamento para colaborar com pesquisas institucionais ao longo do ano, apropriando os dados de seus respectivos *campi*.

## REFERÊNCIAS

AMAZON. **O que é carregamento de dados?**

Disponível em: <https://aws.amazon.com/pt/what-is/etl/>. Acesso em: 30 jan. 2023.

BAELDUNG. **Envio de E-mails com o Spring**. 29 set. 2023.

Disponível em: <https://www.baeldung.com/spring-email>. Acesso em: 02 fev. 2023.

BALSAMIQ. **Quick and Easy Wireframing Tool**.

Disponível em: <https://balsamiq.com/wireframes/>. Acesso em: 30 mar. 2023.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML Guia do usuário**: O mais avançado tutorial sobre Unified Modeling Language (UML), elaborado pelos próprios criadores da linguagem. 2. ed. Rio de Janeiro: Elsevier, 2012.

BUXTON, B. **Sketching User Experiences: Getting the Design Right and the Right Design**. 1ª ed. Amsterdam: Morgan Kaufmann Publishers, 2007.

GITHUB. **Documentação de introdução ao GitHub**.

Disponível em: <https://docs.github.com/pt/get-started>. Acesso em: 30 abr. 2023.

IBM. **O que é ETL (Extrair, Transformar e Carregar)?** 2023a.

Disponível em: <https://www.ibm.com/br-pt/cloud/learn/etl>. Acesso em: 30 jan. 2023.

\_\_\_\_\_. **O que é arquitetura de três camadas (tiers)**. 2023b.

Disponível em: <https://www.ibm.com/br-pt/topics/three-tier-architecture>. Acesso em: 30 jan. 2023.

IFPB. **Sobre o SUAP**. 13 de jul. 2016.

Disponível em: <https://www.ifpb.edu.br/ti/sistemas/servicos/suap/sobre-o-suap>. Acesso em: 04 mar. 2023.

IFPE. **ARIA**. Disponível em: <https://aria.ifpe.edu.br>. Acesso em: 25 out. 2022.

ISO/IEC. **ISO/IEC 9126-1:2001**: Software engineering — Product quality — Part 1: Quality model. Disponível em: <https://www.iso.org/standard/22749.html>. Acesso em: 18 ago. 2023.

\_\_\_\_\_. **ISO/IEC 25010:2011**: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Disponível em: <https://www.iso.org/standard/35733.html>. Acesso em: 18 ago. 2023.

JUNIT TEAM. **Junit 5 User Guide**.

Disponível em: <https://junit.org/junit5/docs/current/user-guide/>. Acesso em: 01 jun. 2023.

MAURYA, A. Why Lean Canvas vs Business Model Canvas?. **Leanstack Blog**, 27 fev. 2012. Disponível em: <https://blog.leanstack.com/why-lean-canvas-vs-business-model-canvas/>. Acesso em: 30 jan. 2023.

MEC. Ministério da Educação. Portaria nº 46, de 10 de janeiro de 2005. **Diário Oficial da União**, Brasília, DF, n. 7, p. 4, 11 jan. 2005. Seção 1.

\_\_\_\_\_. Ministério da Educação. Portaria Normativa nº 40, de 29 de dezembro de 2010. **Diário Oficial da União**, Brasília, DF, n. 249, p. 23-31, 29 dez. 2010, Seção 1.

\_\_\_\_\_. Ministério da Educação, Secretaria de Educação Profissional e Tecnológica (SETEC). **Manual do Usuário: SISTEC - Sistema Nacional de Informações da Educação Profissional e Tecnológica**. 2018. Disponível em: [http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=101781-manual-sistec&category\\_slug=novembro-2018-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=101781-manual-sistec&category_slug=novembro-2018-pdf&Itemid=30192). Acesso em: 06 jun. 2023.

\_\_\_\_\_. Ministério da Educação. **Plataforma Nilo Peçanha**. Disponível em: <https://www.gov.br/mec/pt-br/pnp>. Acesso em: 25 out. 2023.

MICROSOFT. **ETL (extração, transformação e carregamento)**. 2023a. Disponível em: <https://learn.microsoft.com/pt-br/azure/architecture/data-guide/relational-data/etl>. Acesso em: 30 jan. 2023.

\_\_\_\_\_. **TypeScript is JavaScript with syntax for types**. 2023b. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 02 fev. 2023.

MONGODB. **Introduction to MongoDB**. Disponível em: <https://www.mongodb.com/introduction>. Acesso em: 30 jan. 2023.

SOMMERVILLE, I. **Engenharia de software**. 9ª ed. São Paulo: Pearson Prentice Hall, 2011.

ORACLE. **O que é ETL?**. 2023. Disponível em: <https://www.oracle.com/br/integration/what-is-etl/>. Acesso em: 30 jan. 2023.

POSTGRESQL. **PostgreSQL 15**. Disponível em: <https://www.postgresql.org/files/documentation/pdf/15/postgresql-15-A4.pdf>. Acesso em: 30 jan. 2023.

SCHWABER, K. SHUTERLAND, J. **The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game**. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>. Acesso em: 30 jan. 2023.

SOARES, J. P. Importância dos testes de software na qualidade do sistema. **Treinaweb**, 2020. Disponível em: <https://www.treinaweb.com.br/blog/importancia-dos-testes-de-software-na-qualidade-do-sistema>. Acesso em: 02 fev. 2023.

SPRING. **Spring Boot Reference Documentation – Starters**. 2023a. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#using.build-systems.starters>. Acesso em: 01 jan. 2023.

SPRING. **What is Spring Boot?**. 2023b. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 08 jan. 2023.

\_\_\_\_\_. **Getting Started | Building a RESTful Web Service**. 2023c. Disponível em: <https://spring.io/guides/gs/rest-service/>. Acesso em: 08 jan. 2023.

\_\_\_\_\_. **Spring Security**. 2023d. Disponível em: <https://docs.spring.io/spring-security/reference/index.html>. Acesso em: 01 jan. 2023.

\_\_\_\_\_. **Spring Batch Reference Documentation**. 2023e. Disponível em: <https://docs.spring.io/spring-batch/docs/current/reference/html/index.html>. Acesso em: 09 mar. 2023.

\_\_\_\_\_. **Spring Batch**. 2023f. Disponível em: <https://spring.io/projects/spring-batch>. Acesso em: 02 fev. 2023.

\_\_\_\_\_. **Configuring and Running a Job**. 2023g. Disponível em: <https://docs.spring.io/spring-batch/docs/current/reference/html/job.html>. Acesso em: 02 fev. 2023.

\_\_\_\_\_. **Configuring a Step**. 2023h. Disponível em: <https://docs.spring.io/spring-batch/docs/current/reference/html/step.html>. Acesso em: 02 fev. 2023.

\_\_\_\_\_. **Spring Data**. 2023i. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/html/data.html#data.sql.jpa-and-spring-data>. Acesso em: 02 fev. 2023.

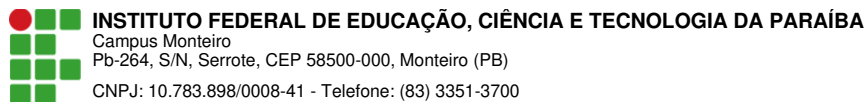
\_\_\_\_\_. **Testing**. 2023j. Disponível em: <https://docs.spring.io/spring-boot/docs/1.5.7.RELEASE/reference/html/boot-features-testing.html>. Acesso em: 02 fev. 2023.

UFPB. **SIGAA**. 22 fev. 2022. Disponível em: <https://proex.ufpb.br/proex/contents/paginas/sigaa>. Acesso em: 4 set. 2023.

UFRJ. **O Pesquisador Institucional**. Disponível em: <https://npi.pr1.ufrj.br/index.php/deveres-do-pi>. Acesso em: 22 ago. 2023.

VERCEL. **Docs | Next.js**. 2023. Disponível em: <https://nextjs.org/docs>. Acesso em: 01 set. 2023.





## Documento Digitalizado Restrito

### Entrega de TCC - REENTREGA CORREÇÕES

**Assunto:** Entrega de TCC - REENTREGA CORREÇÕES  
**Assinado por:** Bruno Santos  
**Tipo do Documento:** Anexo  
**Situação:** Finalizado  
**Nível de Acesso:** Restrito  
**Hipótese Legal:** Informação Pessoal (Art. 31 da Lei no 12.527/2011)  
**Tipo do Conferência:** Cópia Simples

Documento assinado eletronicamente por:

- **Bruno Bezerra dos Santos, ALUNO (201815020013) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - MONTEIRO**, em 25/10/2023 22:52:06.

Este documento foi armazenado no SUAP em 25/10/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 979610  
Código de Autenticação: 7d93ea37a3

