

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Leonardo Brendo Gomes Nascimento

**PIXEL, Plataforma para Integração de Experimentos de
Interoperabilidade em Sistemas Legados de Saúde Pública**

Fortaleza – CE

18 de novembro de 2020

Leonardo Brendo Gomes Nascimento

PIXEL, Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados de Saúde Pública

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Informática Aplicada à Saúde

Orientador: Prof. Dr. Antonio Mauro Barbosa de Oliveira

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

18 de novembro de 2020

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

G633p Gomes, Leonardo.
PIXEL, Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados de
Saúde Pública / Leonardo Gomes. - 2020.
80 f. : il. color.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus
Fortaleza, 2020.

Orientação: Prof. Dr. Antonio Mauro Barbosa de Oliveira.

1. Interoperabilidade. 2. Sistemas de Informação na Saúde. 3. Sistemas Legados. 4. GISSA. I. Título.
CDD 004



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
Av. 13 de maio, 2081 - Bairro Benfica - CEP 60040-531 - Fortaleza - CE - www.ifce.edu.br

LEONARDO BRENDON GOMES NASCIMENTO

PIXEL, Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados de Saúde Pública.

Dissertação submetida à Coordenação do Curso de Pós-graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para a obtenção do título de mestre em Ciência da Computação, área de concentração Sistemas de Computação.

Aprovada em 01/10/2020.

BANCA EXAMINADORA

Prof. Dr. Antonio Mauro Barbosa de Oliveira (Orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Prof. Dr. Antonio Wendell de Oliveira Rodrigues
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Prof. Dr. Luiz Odorico Monteiro de Andrade
Universidade Federal do Ceará - UFC

Prof. Dr. César Olavo de Moura Filho
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE



Ensino Básico, Técnico e Tecnológico, em 02/10/2020, às 12:25, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Cesar Olavo de Moura Filho, Professor do Ensino Básico, Técnico e Tecnológico**, em 02/10/2020, às 13:15, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Antonio Wendell de Oliveira Rodrigues, Professor do Ensino Básico, Técnico e Tecnológico**, em 05/10/2020, às 13:59, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luiz Odorico Monteiro de Andrade, Usuário Externo**, em 14/10/2020, às 14:40, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade do documento pode ser conferida no site https://sei.ifce.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 informando o código verificador **2023172** e o código CRC **98F720E9**.

*Este trabalho dedico à minha esposa Kamila Gomes, à minha mãe Silvana Gomes, ao
meu irmão Leandro Gomes,
aos meus amigos Paulo Castro e Josué Oliveira e à comunidade científica internacional de
interoperabilidade em sistemas de saúde.*

Agradecimentos

Aquele que sempre me concede sabedoria e força para lutar pelos meus sonhos, que sempre esteve e está comigo. Pelo dom da vida e as minhas atribuições que sou demérito, ao meu Deus por mais um trabalho realizado.

À minha esposa Kamila Gomes por sempre me apoiar nos meus estudos e sendo continuamente uma sustentação em minha vida ao que diz respeito ao foco nos meus objetivos, portanto, estando comigo nos momentos tristes e felizes, fazendo transformar a minha fraqueza numa força que eu desconhecia, até então.

À minha mãe Silvana Gomes e ao meu irmão Leandro Gomes por sempre me motivarem na busca pelo progresso nos estudos.

Ao meu orientador e amigo Professor Dr. Mauro Oliveira, responsável por ter me guiado nesta pesquisa e acreditado em mim, sobretudo com a sua dedicação durante este curso de Mestrado Acadêmico. Meu amigo, porque sempre me deu ricos conselhos para vida e, com isso, buscou o melhor para mim.

Aos Professores Dr. César Olavo pelo seu rigor científico nesta dissertação, Dr. Wendell Rodrigues por elucidar questões técnicas pertinentes à proposta, Ms. Valter Costa por sua dedicação em refinar este trabalho e ao Ms. Fábio Gomes por instruir muitos princípios de RES.

Ao meu grande amigo Paulo Castro que sempre esteve comigo percorrendo sobre a PIXEL em demoradas e profícuas reuniões, entre manhãs, tardes, noites e madrugadas.

Ao meu grande amigo Josué Oliveira que me acompanhou durante esta jornada acadêmica, promovendo momentos de motivação e alegria.

Ao Professor Dr. Odorico Monteiro e à Dra. Professora Ivana Cristina por sempre destacar a importância da saúde digital e a relação da Ciência da Computação com a área médica.

À empresa Avicena e o seu CEO Daniel Barreto por mediar a aplicação da PIXEL no sistema GISSA.

Aos meus amigos Geovani Martins, Carlos Estêvão e aos colegas de mestrado e companheiros do LAR/LAMP/LES, PPGCC- IFCE, assim como todos aqueles que contribuíram de forma significativa, direta e indiretamente com a minha formação e, conseqüentemente, para a conclusão desta etapa da minha vida.

“Não abandones a sabedoria e ela te guardará; ama-a, e ela te protegerá. A sabedoria é a coisa principal; adquiere pois a sabedoria, emprega tudo o que possuis na aquisição de entendimento. Exalta-a, e ela te exaltará; e, abraçando-a tu, ela te honrará.”
(Provérbios de Salomão).

Resumo

Sistemas de Informação em Saúde (SIS) vêm aumentando em complexidade e volume de dados. Estes dados nem sempre seguem um padrão, o que dificulta a integração de serviços e a interoperabilidade entre os sistemas, agravando, também, o desenvolvimento de novas aplicações no contexto da saúde digital. O Ministério da Saúde (MS) vem acompanhando esta tendência internacional ao buscar estabelecer padrões de integração e interoperabilidade a partir da portaria 2.073 de 31 de agosto de 2011. Como resultado dessas iniciativas, adotou-se o padrão *Open Electronic Health Record* (OpenEHR) no Sistema Único de Saúde (SUS) e, mais recentemente, o Conjunto Mínimo de Dados (CMD). Entretanto, a interoperabilidade em saúde não se restringe somente à questão do uso de padrões. Existem outras problemáticas relacionadas a integração e interoperabilidade de sistemas, em âmbito nacional e internacional. Este trabalho realiza um estudo sobre interoperabilidade e agregação de serviços de saúde e propõe e implementa a PIXEL, uma Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados de Saúde Pública que trata os níveis básico, sintático e semântico de interoperabilidade, proposto pela *Health Information and Management Systems Society* (HIMSS). Esta solução foi implementada no GISSA[®], um sistema inteligente que usa *Machine Learning* (ML) e ontologias na geração de *dashboards*, indicadores, análise de risco e alertas para apoio à tomada de decisão para governança em saúde. Em consequência, o GISSA[®] passa a disponibilizar, com a PIXEL, uma *Application Programming Interface* (API) *Representational State Transfer* (REST) na comunicação com outros sistemas, além da interoperabilidade entre diferentes padrões de saúde (OpenEHR, FHIR, etc).

Palavras-chave: Interoperabilidade, Sistemas de Informação na Saúde, Sistemas Legados e GISSA.

Abstract

Health information systems (HIS) are increasing in complexity and volume of data. These data do not always follow a pattern, which leads to a worsening of the integration of services and interoperability of systems, making it difficult to communicate between them. The Ministry of Health (MH) has been following this international trend when seeking to establish standards of integration and interoperability from Ordinance 2,073 of August 31, 2011. As a result of these initiatives, the Open Electronic Health Record (OpenEHR) standard was adopted in the Unified Health Systems (UHS) and, more recently, the Minimum Data Set (MDS). However, the problem of interoperability in health is not restricted to the issue of standards. There are other problems related to the integration and interoperability of systems, both nationally and internationally. This work makes a study on interoperability and aggregation of health services and proposes and implements PIXEL, a Platform for the Integration of Interoperability Experiments in Legacy Public Health Systems that addresses the foundational, syntactic and semantic levels of interoperability, proposed by the Health Information and Management Systems Society (HIMSS) . This solution was implemented in GISSA[®], an intelligent system that uses Machine Machine Learning (ML) and ontologies in the generation of alerts and support to the decision-making of its users. Consequently, GISSA[®] starts to make available, with PIXEL, an Application Programming Interface (API) Representational State Transfer (REST) on communication with other systems, in addition to interoperability between different standards of health (OpenEHR, FHIR, etc.).

Keywords: Interoperability, Health Information Systems, Legacy Systems and GISSA.

Lista de ilustrações

Figura 1 – Ilustração da interoperabilidade do HIMSS.	31
Figura 2 – Categorização de interoperabilidade pelo HIMSS.	32
Figura 3 – SIS Meu digisus.	34
Figura 4 – Menu Inicial Gissa.	36
Figura 5 – Arquitetura do GISSA.	37
Figura 6 – Dashboard risco de óbito materno.	38
Figura 7 – Relatório de risco para óbito materno.	39
Figura 8 – Indicador de taxa de mortalidade específica - idade/ 100 Hab.	39
Figura 9 – Alertas disponíveis no GISSA.	40
Figura 10 – Portal Semântico.	47
Figura 11 – Arquitetura do Portal Semântico.	48
Figura 12 – <i>Framework</i> de integração semântica.	48
Figura 13 – API <i>gateway</i>	50
Figura 14 – Arquitetura do <i>hub</i> de Integração.	52
Figura 15 – Utilização do barramento <i>plugin</i> de serviços.	53
Figura 16 – Etapas para a integração na interoperabilidade de agregação de novos serviços.	53
Figura 17 – Escopo de funcionamento da PIXEL.	57
Figura 18 – Arquitetura da PIXEL.	59
Figura 19 – Diagrama de classe da estrutura Gerador de Modelos.	60
Figura 20 – Trecho do código usado para listar a base de dados.	60
Figura 21 – Trecho do código usado especificar a base de dados.	60
Figura 22 – Trecho do código usado para listar as tabelas disponíveis.	61
Figura 23 – Diagrama de classe da estrutura Modelos.	61
Figura 24 – Código da classe BaseModel.	62
Figura 25 – Código da classe GerarArquivoModel.	62
Figura 26 – Código da classe UrlMappings.	62
Figura 27 – Fluxo de funcionamento da arquitetura PIXEL em BPMN.	63
Figura 28 – Diagrama de pacotes da PIXEL.	66
Figura 29 – Diagrama de classes do pacote <i>MVC</i>	67
Figura 30 – Diagrama de classes do pacote <i>UTIL</i>	68
Figura 31 – Diagrama de pacotes dos sistemas participantes.	69
Figura 32 – Arquitetura GISSA utilizando a PIXEL.	69
Figura 33 – Selecionando a base de dados.	70
Figura 34 – Selecionando a tabela.	70
Figura 35 – Base de dados e tabela selecionadas.	71

Figura 36 – Criação do modelo. 71

Figura 37 – Integração dos Sistemas. 71

Lista de tabelas

Tabela 1 – Comparação dos trabalhos relacionados com os requisitos desejados. . .	45
Tabela 2 – Requisitos Funcionais da PIXEL	56
Tabela 3 – Requisitos Não Funcionais da PIXEL	57

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
AWF	Atlântico Web Framework
BI	<i>Business Intelligence</i>
BPMN	<i>Business Process Model and Notation</i>
CMD	Conjunto Mínimo de Dados
CNES	Cadastro Nacional de Estabelecimento de Saúde
CRUD	<i>Create, Read, Update and Delete</i>
DATASUS	Departamento de informática do Sistema Único de Saúde
DSL	<i>Domain-Specific Language</i>
EE	<i>Enterprise Edition</i>
EHR	<i>Electronic Health Record</i>
e-SUS	Estratégia de Informatização do SUS
ETL	<i>Extract, Transform and Load</i>
FHIR	<i>Fast Healthcare Interoperability Resources</i>
GISSA	Governança Inteligente em sistemas de Saúde
HATEOAS	<i>Hypermedia as the Engine of Application State</i>
HIMSS	<i>Healthcare Information and Management Systems Society</i>
HL7	<i>Health Level 7</i>
HTTP	<i>HyperText Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IS	Integração Semântica
JDBC	<i>Java Database Connectivity</i>

JDK	<i>Java Development Kit</i>
JSON	<i>JavaScript Object Notation</i>
JSP	<i>JavaServer Pages</i>
JSTL	<i>JavaServer Pages Standard Tag Library</i>
LOINC	<i>Logical Observation Identifiers Names and Codes</i>
ML	<i>Machine learning</i>
MDA	<i>Model-Driven Architecture</i>
MS	Ministério da Saúde
MVC	<i>Model, View and Controller</i>
NoSQL	<i>Not Only Structure Query Language</i>
ODBAPI	<i>Unified REST API for Relational and NoSQL Data Stores</i>
ORESTES	<i>Scalable and Unified REST API for Cloud Data Stores</i>
OpenEHR	<i>Open Electronic Health Record</i>
OWL	<i>Ontology Web Language</i>
PEC	Prontuários Eletrônico do Cidadão
PEP	Prontuários Eletrônico do Paciente
PNIS	Política Nacional de Informação e Informática em Saúde
PoC	Prova de Conceito
REST	<i>Representational State Transfer</i>
SIM	Sistema de Informação de Mortalidade
SINAN	Sistema de Informação de Agravos de Notificação
SINASC	Sistema Nacional de Nascidos Vivos
SI-PNI	Sistema de Informação do Programa Nacional de Imunização
SIS	Sistemas de Informação em Saúde
SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>

SRS	<i>Software Requirements Specification</i>
SUS	Sistema Único de Saúde
TXT	Texto
TISS	Troca de Informações em Saúde Suplementar
TI	Tecnologia da Informação
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
WS	Web Semântica
WSDL	<i>Web Service Description Language</i>
XML	<i>Extensible Markup Language</i>

Sumário

1	INTRODUÇÃO	25
1.1	Motivação para a Dissertação	27
1.2	Descrição do problema	28
1.3	Objetivos	29
1.3.1	Objetivo Geral	29
1.3.2	Objetivos Específicos	29
1.4	Produção científica	29
1.5	Estrutura da Dissertação	30
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	Interoperabilidade	31
2.2	Sistemas de Informação na Saúde (SIS)	33
2.3	Sistemas legados	35
2.4	GISSA	36
3	TRABALHOS RELACIONADOS	41
3.1	Disponibilizando Informações via <i>API REST</i> com Modelos	41
3.2	Plataforma Interoperável em sistemas de informação	42
3.3	Interoperabilidade em Sistemas de informação na Saúde	43
3.4	Quadro Comparativo	44
4	ESTUDO DE INTEROPERABILIDADE DO SISTEMA GISSA	47
4.1	Portal semântico no GISSA	47
4.2	RES no GISSA	50
4.3	Agregação de novos serviços no GISSA	52
5	PLATAFORMA PIXEL	55
5.1	Cenário de Aplicação	55
5.2	Requisitos funcionais e requisitos não-funcionais	56
5.3	Arquitetura PIXEL	58
5.4	Fluxo de Funcionamento	63
5.5	Tratamento da Interoperabilidade	64
5.6	Aspectos de implementação da PIXEL	64
5.6.1	Funcionamento tecnológico da PIXEL	64
5.6.2	Tecnologias e Modelagem da PIXEL	65
6	APLICANDO A PLATAFORMA PIXEL EM AMBIENTE REAL	69

7	CONCLUSÃO	73
7.1	Limitações do Trabalho	73
7.2	Trabalhos Futuros	74
	REFERÊNCIAS	75

1 Introdução

A diversidade dos dados e informações fornecidas à administração pública traz grandes desafios para cooperação e à formalização de processos. A grande disseminação das informações sem um planejamento apropriado, em virtude das muitas fontes de dados, tem gerado dificuldades cada vez maiores para a integração das informações nas instâncias governamentais e privadas (RIBEIRO, 2011).

Governos falham em implementar interoperabilidade em seus sistemas públicos de saúde, por diversas razões, dentre elas o não cumprimento de normas fundamentais por parte dos componentes de uma rede de colaboração (HENNING, 2013). Isso impacta diretamente nos diferentes setores governamentais que têm a necessidade de integrar informações aos seus sistemas. De acordo com (MAHMOUD, 2013), a integração de sistemas não deve ter uma solução única em determinado momento, sendo uma obrigação que os governos e organizações deverão satisfazer no futuro para os setores que utilizam tecnologias.

No setor de saúde, as entidades geram diariamente um denso volume de informações em formatos variados. Para agravar a situação, é comum que entidades adotem diferentes sistemas para tratar seus dados. Uma das prioridades destacadas por estas instituições está diretamente ligada à integração destes sistemas com os seus parceiros internos e externos, assegurando a troca de informação (FARINELLI; ALMEIDA, 2014). A integração de Sistemas de Informação de Saúde (SIS) é uma característica fundamental do conceito interoperabilidade de sistemas, objeto deste trabalho. Assim, a complexidade das aplicações futuras na área de saúde digital exigem SIS interoperáveis, uma tendência mundial.

Considerando a necessidade de assegurar ao cidadão o registro dos dados relacionados à saúde e a interoperabilidade tecnológica dos serviços nos distintos níveis da Federação para estabelecer o intercâmbio das informações e a celeridade dos processos à nível local e nacional, o governo brasileiro, sensível a esta tendência e a necessidade da implantação efetiva da interoperabilidade em seus SIS, vem tentando estabelecer normas para mitigar problemas de integração, assim como a regulamentação do uso de padrões de interoperabilidade e informação em saúde para SIS na perspectiva do SUS, nas esferas Municipais, Distritais, Estaduais e Federal, e para os sistemas de iniciativas privadas e do setor de saúde suplementar.

A interoperabilidade tornou-se uma preocupação evidente no Brasil a partir da portaria 2.073¹ de 31 de agosto de 2011 do Ministério da Saúde e, mais recentemente a adoção do CMD (SAÚDE, 2015). Um dos capítulos desta portaria trata da definição

¹ http://bvsms.saude.gov.br/bvs/saudelegis/gm/2011/prt2073_31_08_2011.htm

e adoção do padrão de interoperabilidade de informações de Saúde utilizando o padrão OpenEHR, incluindo a definição de detalhes e os principais atributos dos serviços, seus componentes, atividades e políticas necessárias (MORAIS; GOMES; COSTA, 2014).

Por meio desta portaria, o Ministério da Saúde estabeleceu um catálogo de serviços e um catálogo de padrões de informações com objetivo de viabilizar a interoperabilidade em seus SIS, tais como, tecnologia *Web Service*, no padrão *Simple Object Access Protocol* (SOAP) 1.1 ou superior; *WS-Security* para criptografia e assinatura digital das informações. Os *Web Services* têm sua identificação via *Uniform Resource Identifier* (URI) e são descritos e definidos pela *Web Service Description Language* (WSDL). Sistemas Legados devem ser encapsulados em padrões *eXtensible Markup Language* (XML). Para a definição do Registro Eletrônico em Saúde (RES) foi utilizado o OpenEHR. Foi usado na integração dos resultados e solicitações de exames, o padrão *Health Level 7* (HL7) e utilizada a terminologia SNOMED-CT para codificação de termos clínicos e mapeamento das terminologias. A seguir são listados os padrões utilizados na portaria:

- Os padrões de Troca de Informações em Saúde Suplementar (TISS) foram usados nos sistemas de saúde suplementar;
- O padrão HL7 CDA foi utilizado na definição da arquitetura do documento clínico;
- O padrão DICOM foi implementado para a representação da informação relativa a exames de imagem;
- O padrão *Logical Observation Identifiers Names and Codes* (LOINC) foi implementado para a codificação de exames laboratoriais;
- A ISBT 128 foi usada para a codificação de dados de identificação das etiquetas de produtos;
- A ISO 13606-2 foi incorporada para a interoperabilidade de modelos de conhecimento;
- A especificação de integração IHE-PIX (*Patient Identifier Cross-Referencing*) foi usada para estabelecer o cruzamento de identificadores de pacientes de diferentes sistemas de informação.

Para o *Institute of Electrical and Electronics Engineers* (IEEE)², a interoperabilidade é a capacidade de um sistema ou produto de funcionar com outros sistemas ou produtos, sem nenhum tipo de esforço especial por parte de um cliente (JAY; GOETZ, 1988). Segundo o IEEE, a aplicação da interoperabilidade é possível somente a partir da utilização rigorosa das normas e padrões estabelecidos. Portanto, a interoperabilidade está diretamente relacionada à cooperação definida por regras específicas, padrões e políticas que possibilitam o intercâmbio integrado das informações.

Em outra definição, a *Healthcare Information and Management Systems Society*

² <https://www.ieee.org/>

(HIMSS)³ (PRESS, 2010) conceitua interoperabilidade como a capacidade de diferentes sistemas de informação e aplicativos de se comunicarem, trocarem dados e usá-los de forma transparente, assim como as informações associadas a integração. Com foco na área da saúde, HIMSS propõe níveis de interoperabilidades na tentativa de solucionar os problemas de integração em sistemas de informação.

Assim, além da questão da interoperabilidade em padrões de informações na saúde e de bases heterogêneas, existem outras problemáticas, como a que envolve a agregação de novos serviços em plataformas legadas ou de tecnologias proprietárias. Ao se tentar comunicar sistemas legados, tem-se uma acentuada dificuldade. Este fato é agravado pela escassa documentação trazendo informações de projeto, dificultando a manutenção e, especialmente, a evolução destes sistemas.

1.1 Motivação para a Dissertação

A problemática abordada neste trabalho é inspirada nas discussões de interoperabilidade em sistemas públicos de saúde, bem como na integração de informações e serviços em diferentes SIS. A interoperabilidade é um desafio em sistemas de informação criados por diferentes equipes há décadas e só tem se agravado à medida em que a evolução dos sistemas digitais acontecem. Desta forma, os SIS não estão isentos tendo dificuldades inerentes, tais como padrões de informações, identificação em termos de terminologias médicas, entre outros. Em geral, a interoperabilidade é tratada isoladamente e de forma singular em SIS.

Ao fazer um levantamento sobre interoperabilidade por meio de plataformas na área da saúde, nota-se que poucos trabalhos propõem soluções para mais de um tipo de interoperabilidade. Assim, a principal motivação neste trabalho é discutir a importância e propor soluções no aspecto de interoperabilidade que, em geral, não é levada em consideração no planejamento e operação em sistema público de saúde, sejam privados ou governamentais.

Ao elencar categorias de interoperabilidades encontradas na literatura, percebe-se que a integração de SIS está cada vez mais complexa, o que leva à repetição desnecessária de procedimentos quando o usuário é atendido em diferentes locais, tendo como consequência o maior custo e menor eficiência tanto para o provedor de saúde quanto para o usuário. A expectativa deste trabalho é contribuir para o melhor entendimento da importância do conceito de interoperabilidade no planejamento e operação dos SIS, bem como na solução de problemas que surgem no seguinte domínio de interoperabilidade aqui tratado: a agregação de novos serviços em sistemas legados ou com tecnologias proprietárias.

³ <https://www.himss.org/>

1.2 Descrição do problema

Considere dois cenários do cotidiano onde um usuário de um sistema governamental de saúde precisa fazer uso de suas informações em outro local do qual ele normalmente é atendido, seja ele um posto de saúde ou um hospital. Idem para um cenário privado, onde o usuário é atendido por diversos profissionais do mesmo plano de saúde. Nos dois casos o usuário é, em geral, clinicamente desconhecido, o que pode resultar no retrabalho de realizar novos exames ou consultas e, como consequência, a demora dos diagnósticos e tratamentos, incorrendo em custos desnecessários. Outrossim, informações valiosas acerca da saúde do indivíduo são subutilizadas e as instituições, tanto públicas como privadas, perdem a oportunidade de promover melhorias no atendimento por meio de estatísticas ou mesmo aplicando técnicas modernas de inteligência artificial nos processos de governança dos sistemas de saúde.

Estes dois cenários ilustram vários problemas da falta de interoperabilidade entre sistemas de saúde que vão desde os aspectos semânticos dos diferentes padrões adotados nos RES, nos Prontuários Eletrônico do Paciente (PEP) ou mesmo nos Prontuários Eletrônico do Cidadão (PEC), quando existem, incompatibilidades sintáticas, até a questão de comunicação destes sistemas. Esse problema se agrava quando neste cenário são envolvidos os chamados sistemas legados, estruturas proprietárias nem sempre disponíveis e bem documentadas.

Assim, à medida em que os sistemas são interoperáveis abrem-se caminhos para o tratamento das informações nos aspectos de comunicação, sintático e semântico dentro de uma estrutura de atendimento, governamental ou privada, possibilitando sistemas mais eficientes pois o usuário será acompanhado em qualquer lugar que procure atendimento. Desta forma, incorrerá em grandes avanços na integração de sistemas, promovendo a comunicação entre os mesmos, fazendo que os usuários destes sistemas deixem de ser clinicamente desconhecidos e possam trocar suas informações.

Este trabalho realiza um estudo qualitativo sobre o conceito de interoperabilidade em SIS e propõe e implementa a Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados (PIXEL), uma solução que busca mitigar as questões de interoperabilidades de comunicação, sintática e semântica para o problema acima descrito e de integração de novos serviços em SIS, legados ou de iniciativa proprietária.

A Prova de Conceito da proposta PIXEL foi realizada na plataforma Governança Inteligente em Serviços de Saúde (GISSA) (OLIVEIRA et al., 2010; LOPES; VIDAL; OLIVEIRA, 2016; SILVA et al., 2017; ANDRADE; OLIVEIRA; RAMOS, 2015), um sistema inteligente para tomada de decisão em saúde. Foi realizada a integração de duas instâncias do GISSA que simulam a interoperabilidade de SIS e a integração de novos serviços em sistemas de iniciativa proprietária.

1.3 Objetivos

1.3.1 Objetivo Geral

Propor, especificar e implementar a PIXEL, um modelo de interoperabilidade nos níveis fundamental (comunicação), sintática e semântica, no contexto definido pela HIMSS, que solucione o problema de integração de serviços em sistemas legados e de tecnologia proprietária em Sistemas de Informação de Saúde.

1.3.2 Objetivos Específicos

- Realizar uma pesquisa qualitativa sobre o estado da arte da interoperabilidade de comunicação, sintática e semântica em SIS, no contexto definido pela HIMSS, com foco na solução de integração de serviços em sistemas legados ou de tecnologia proprietária, tanto nos sistemas públicos e privados brasileiros;
- Elencar requisitos de soluções existentes de interoperabilidade, a partir de um extenso trabalho comparativo, com a finalidade de analisar e identificar soluções automatizadas e adequadas a resolver problemas de integração de serviços em sistemas legados e de tecnologia proprietária ;
- Especificar a PIXEL, uma solução arquitetural para problemas de interoperabilidades de comunicação, sintática e semântica em SIS, considerando diferentes padrões de RES, notadamente o FHIR e o OpenEHR;
- Implementar a plataforma PIXEL, em conformidade com a solução arquitetural proposta;
- Aplicar a PIXEL no ambiente GISSA, uma plataforma de Governança Inteligente em Sistemas de Saúde, utilizando-a como Prova de Conceito (PoC) da agregação de novos serviços em sistemas legados ou de iniciativa proprietária.

1.4 Produção científica

Durante este projeto do mestrado acadêmico, os seguintes trabalhos foram elaborados e aceitos para publicação:

- Leonardo Nascimento, Paulo Castro, Mauro Oliveira, Fabio Jose, Raimundo Valter, César Moura, Renato Freitas, Odorico Monteiro; “PIXEL, Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados de Saúde Pública”, VIII Escola Regional de Computação Ceará, Maranhão, Piauí (ERCEMAPI 2020);
- Vinícius N. Barbosa, Helen Abdala R. Ferreira, Leonardo Gomes, Fábio Gomes, Ivana Barreto, Odorico Monteiro e Mauro Oliveira; ”*SmartRES* - Uma plataforma

IoT para Monitoramento Inteligente em Saúde e sua Aplicação no Contexto da COVID-19”, 20° Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS), 2020;

- Leonardo Nascimento, Renato Freitas, César Olavo, Ivana Holanda, Odorico Monteiro and Mauro Oliveira; “*Analysis of Interoperability in Public Health Systems*”, 8th International Workshop on ADVANCES in ICT Infrastructures and Services, Jan, 2020;
- Carlos E. B. Sousa, Leonardo B. G. Nascimento, Cláudio. M. S. Medeiros, Mauro Oliveira and Iago A. Trajano; “*Decision Support Software for Melanoma Skin Cancer Detection (DECIME)*”, SP4HC: ECAI 2020 Workshop on Singular Problems for Healthcare;

1.5 Estrutura da Dissertação

O restante do texto da dissertação está organizado em seis capítulos:

No capítulo 2, a Fundamentação Teórica apresenta os principais conceitos usados nesta pesquisa. Inicialmente, o conceito de Interoperabilidade é apresentado considerando-se a definição usada pela HIMSS para a área da saúde. Em um segundo momento, os SIS são apresentados dando-se ênfase ao alcance de suas funcionalidades na perspectiva da saúde. Em seguida, no contexto dos sistemas de saúde digital, os sistemas legados são discutidos, evidenciando-se a complexidade de mantê-los ativos eficazmente. Finalmente, tem-se o GISSA, um sistema de saúde com muitos cenários de interoperabilidade, usado para a realização da Prova de Conceito (PoC) da PIXEL.

O capítulo 3 descreve o estado da arte para disponibilização de informações, via *API REST*, usando a ideia de Modelos, Plataforma Interoperável em sistemas de informação e Interoperabilidade em SIS.

O capítulo 4 apresenta um estudo sobre Interoperabilidade, destacando três iniciativas de soluções: Portal semântico e o RES no GISSA, além da agregação de novos serviços no GISSA.

O capítulo 5 discorre sobre a plataforma proposta, com foco no cenário de aplicação, requisitos funcionais e requisitos não funcionais, arquitetura, tratamento de interoperabilidades e aspectos de implementação da PIXEL.

O capítulo 6 apresenta a PoC que foi implementada como intuito de mostrar a aplicabilidade da PIXEL no GISSA, assim como a exibição e integração das informações e os resultados dos experimentos realizados.

O capítulo 7 apresenta considerações finais, limitações do trabalho e trabalhos futuros.

2 Fundamentação Teórica

Nesta seção são pormenorizadas as definições fundamentais utilizadas nesse trabalho: Interoperabilidade, Sistemas de Informação na Saúde (SIS), Sistemas Legados e GISSA.

2.1 Interoperabilidade

Em Ciência da Computação, Interoperabilidade é a capacidade que um sistema possui de compartilhar, comunicar e trocar aplicações e informações com outros sistemas que possuem estruturas e dados díspares (BISHR, 1997; SAYÃO; MARCONDES, 2008; SHETH, 1999). O termo Interoperabilidade está relacionado à capacidade de utilizar, por parte do processo ou sistema, as informações de outro sistema por meio de padrões estabelecidos (VERNADAT, 2007).

Para (SANTOS, 2008), Interoperabilidade é definida como a qualidade capaz de fazer um sistema ou processo usar as informações e/ou as funções de outro sistema ou processo através da adesão a padrões comuns. Na visão de (MILLER, 2000), tem-se o entendimento de Interoperabilidade como um processo de caráter contínuo centralizado em assegurar que os sistemas ou processos troquem informações. Segundo (SANTOS, 2011), Interoperabilidade é a competência que máquinas e sistemas têm de trocar, processar e compreender corretamente as informações, sendo requisito indispensável para a entrega flexível e a viabilização eletrônica de serviços governamentais.

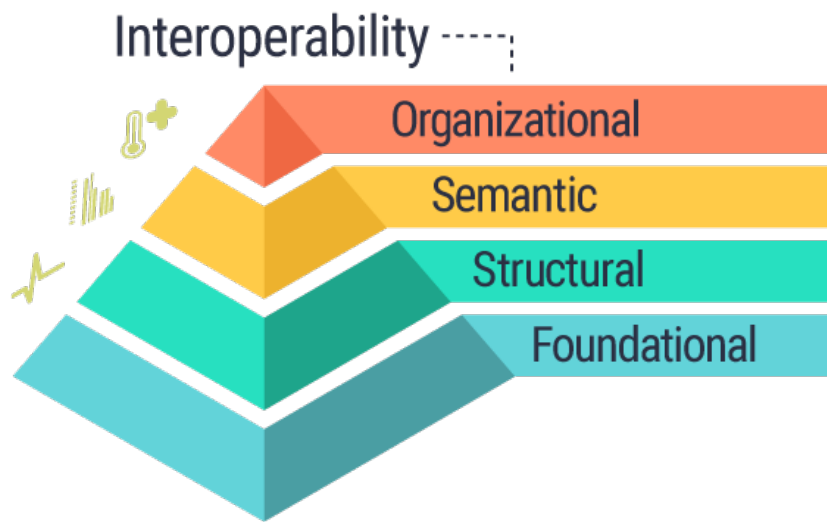
Levando em consideração a dimensão da abrangência do termo interoperabilidade e seu significado. A HIMSS fortifica o termo, ao fazer uma nova proposta de definição de interoperabilidade. As Figuras 1 e 2 ilustram o conceito de Interoperabilidade da HIMSS, apresentado a seguir:

Figura 1 – Ilustração da interoperabilidade do HIMSS.



Fonte: Site HIMSS.

Figura 2 – Categorização de interoperabilidade pelo HIMSS.



Fonte: Site HIMSS.

1. **Básica (nível 1):** Estabelece os requisitos de interconectividade necessários para que um sistema/aplicativo comunique com segurança e receba dados de outro sistema e, assim, seja estabelecido a conexão;
2. **Estrutural (Nível 2):** Define o formato, sintaxe e organização da troca de dados, inclusive no nível do campo de dados para interpretação;
3. **Semântica (Nível 3):** Fornece modelos subjacentes comuns e codificação dos dados, incluindo o uso de elementos de dados com definições padronizadas de conjuntos de valores e vocabulários de codificação publicamente disponíveis;
4. **Novo organizacional (nível 4):** Inclui considerações de governança, políticas, sociais, legais e organizacionais para facilitar a comunicação segura, contínua e oportuna e o uso de dados nas organizações, entidades e indivíduos.

Ao tratar-se dos tipos de interoperabilidades, surgem distintas problemáticas relacionadas às tipificações. Nesse aspecto, (SHETH, 1999) indica que os problemas atrelados a cada tipificação podem apresentar diferenças sintáticas, semânticas e estruturais por estarem relacionados a fontes de informações manipuladas pelo sistema. Para atenuar os problemas oriundos dos tipos de interoperabilidades, (SHETH, 1999) sugere que o grau de cooperação entre sistemas deve ser categorizado em pelo menos três níveis de acordo com a cooperação estabelecida:

1. **Cooperação Técnica:** Este tipo de cooperação busca promover a interoperabilidade tecnológica ao contexto;

2. **Cooperação de Conteúdo:** Este tipo de cooperação busca promover interoperabilidade semântica ao contexto;
3. **Cooperação Organizacional:** Este tipo de cooperação está relacionado à Interoperabilidade organizacional.

De acordo com os níveis de cooperação sugeridos por (ARMS et al., 2002), os autores do trabalho (SAYÃO; MARCONDES, 2008) complementam os níveis de cooperação adicionando questões políticas que objetivam estabelecer grupos de discussão com finalidade de padronizar as diretrizes e políticas fundamentais. Nesse sentido, o Brasil tem aberto discussão no âmbito da Política Nacional de Informação e Informática em Saúde (PNIIS) com vistas a promover interoperabilidade entre os diversos sistemas do governo brasileiro.

Esse tema é discutido na portaria 2.073 de 2011 do MS que, em um dos seus capítulos, traz uma abordagem política para fomentar a iniciativa de padronização e uso de interoperabilidade em SIS. Uma das sugestões dessa portaria é a aplicação de ontologias e terminologias para tratar as questões de interoperabilidade.

O termo interoperabilidade permeia todo escopo deste trabalho, sendo indispensável para a realização do estudo que alcança os níveis propostos por HIMSS, oportunizando a concepção da plataforma PIXEL, proposta focada na integração dos SIS.

2.2 Sistemas de Informação na Saúde (SIS)

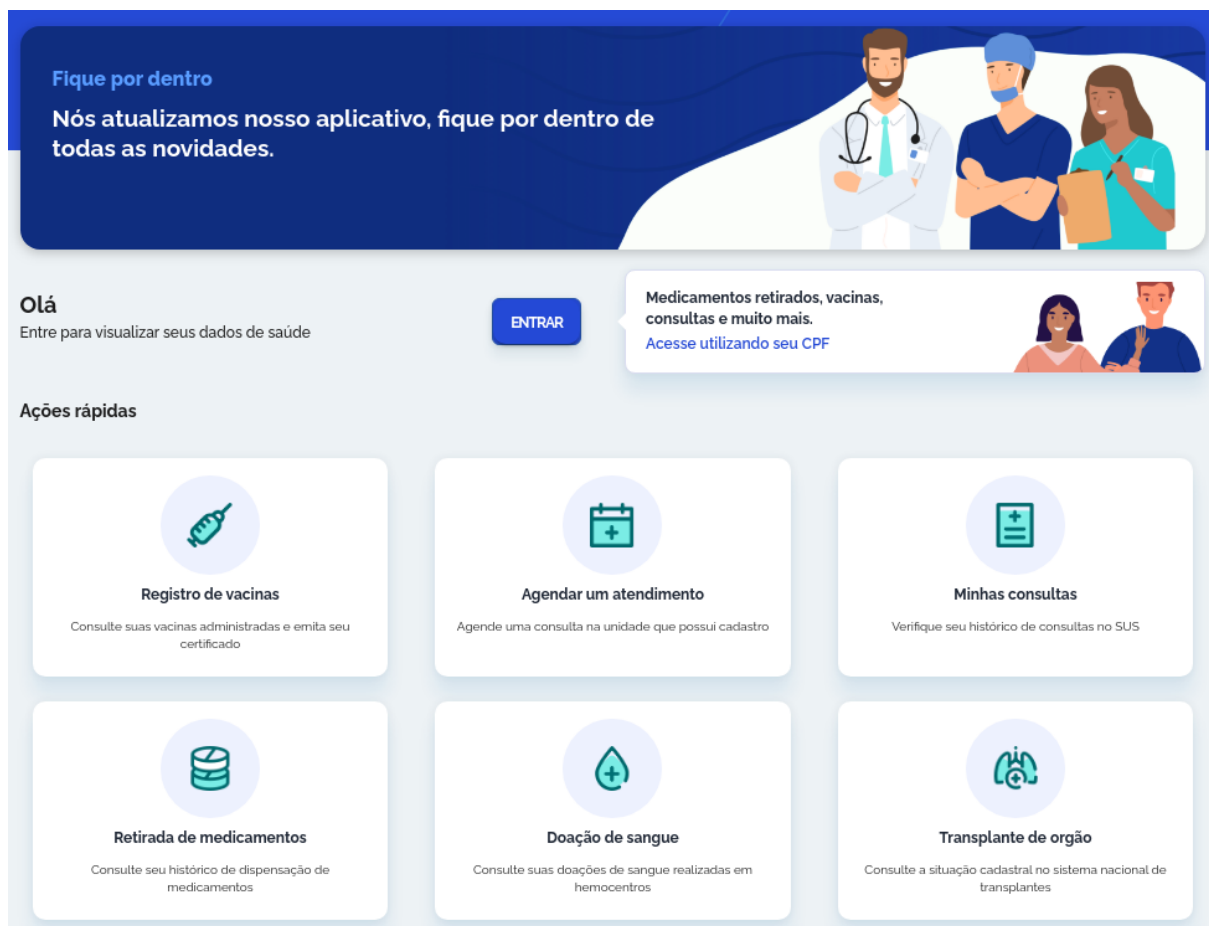
Atualmente, as organizações utilizam sistemas de informação com a finalidade de auxiliar no cumprimento de suas missões (AMARAL, 1994), desempenhando um papel indispensável na gestão da informação. O setor da saúde vem se destacando na intensidade e complexidade de suas atividades. Há imprecisão nos conceitos, atualizações recorrentes do estado de saúde dos pacientes, uma grande variabilidade estrutural e organizacional dos processos, uma extensa quantidade de profissionais com distintas qualificações e perfil, assim como as diferentes culturas inerentes de cada organização. Estes motivos associados a complexidade do *workflow* dos processos clínicos e dos dados neles contidos representam riscos para o desenvolvimento, implementação e manutenção de software para este mercado (CHU, 2005).

A promissora revolução na indústria de Tecnologias da Informação (TI) e a modernização dos processos humanos tem gerado mudanças significativas na maneira de utilizar os sistemas de informação (ZHANG; DU; LI, 2011). Esta realidade está impactando também as organizações e serviços de saúde, onde gestores, funcionários, médicos e enfermeiros têm o seu trabalho cada vez mais dependente de sistemas digitais (CRUZ-CORREIA et al., 2012).

Os sistemas feitos para a área da saúde têm demonstrado grande potencial de efficientização dos procedimentos administrativos, clínicos e, conseqüentemente, na evolução da qualidade dos cuidados de saúde (ABUGABAH, 2017). Em contrapartida, os SIS auxiliam a mitigar custos médicos, ocorrência de erros em cirurgias médicas e a política de proteção e sigilo dos dados dos pacientes (ZEINALI; ASOSHEH; SETAREH, 2016).

O SIS pode ser compreendido como uma maneira de viabilizar a aquisição, organização e análise dos dados fundamentais à definição de problemas e riscos para a saúde, avaliação da eficácia, influência e eficiência que os serviços oferecidos consigam ter no estado de saúde da população, contribuindo para a fomentação de conhecimento acerca da saúde e dos assuntos a ela relacionados (WHITE, 1980). A Figura 3 apresenta um exemplo de SIS chamado Meu digisus. Nele podem ser observados várias funcionalidades típicas de SIS.

Figura 3 – SIS Meu digisus.



Fonte: Site Meu digisus.

SIS têm crescido em complexidade e vêm gerando desafios para integrar informações, em especial ao se tratar de sistemas com iniciativas proprietárias ou legados.

2.3 Sistemas legados

Em diversas empresas, softwares antigos ainda são indispensáveis. Empresas tornam-se dependentes destes sistemas e qualquer falha pode comprometer a continuidade das operações da empresa, dado a dificuldade de manutenção. A esses sistemas, em geral antigos e de grande duração, é atribuído o nome de Sistemas Legados (FILHO, 2003). Por serem fundamentais em muitas organizações, (JOHANN, 2016) enfatiza que sistemas legados são cruciais para o negócio pois geram receita, fazendo parte de processos chave do negócio.

Sistemas legados são, portanto, críticos para o negócio da empresa. Implementado há anos e tendo consolidado grandes alterações nas suas execuções rotineiras, geralmente possuem documentação desatualizada, obscura ou inexistente, sofrendo manutenção por várias pessoas que participaram da equipe durante anos. Desse modo, são softwares de suma importância para o bom funcionamento e regularidade dos processos organizacionais da empresa.

De acordo com (LIMA et al., 2016), pode-se definir Sistemas Legados como sistemas sociotécnicos baseados em computadores desenvolvidos no passado, frequentemente usando tecnologias mais antigas ou obsoletas. Esses sistemas incluem não apenas hardware e software, mas, também, processos e procedimentos legados, velhas formas de fazer coisas que dificilmente são mudadas porque estão baseadas nestes softwares. As mudanças em uma parte do sistema envolvem, inevitavelmente, mudanças em outros componentes.

Sistema legado é qualquer sistema de informação que resiste significativamente à alteração e à evolução tecnológica, porém mantém seu valor como produto (BRODIE; STONEBRAKER, 1995) ou um software que possui criticidade nas atividades para os negócios do dia-a-dia da empresa, resistindo às modificações (BAKAR; RAZALI; JAMBARI, 2019). Para (UMAR, 1997), sistemas legados são softwares com valoração crítica para o negócio em produção das empresas a partir de cinco anos de utilização. Nesse sentido, sobrevivem ao longo do tempo nas empresas e instituições, por manter um grande significado para a manutenção do empreendimento, porém, as tecnologias usadas estão depreciadas ou obsoletas, tornando cada vez mais difícil a sua manutenção e sobrevivência.

O conceito de Sistema Legado é importante para este trabalho, uma vez que este tipo de sistema em geral tem grandes dificuldades para se integrar e “conversar” com outros sistemas, ou seja, tem problemas de interoperabilidade.

Finalmente, é importante não confundir Sistema Legado com sistema com tecnologia proprietária que, por diversas razões, optam em seu desenvolvimento por ferramentas e plataformas que não são “padrão” no mercado, incorrendo, por vezes, em problemas semelhantes a dos sistemas legados (documentação pobre, dificuldade de manutenção, pessoal especializado, etc.).

2.4 GISSA

O GISSA é um sistema de Governança Inteligente em Saúde que utiliza o Atlântico Web Framework (AWF), desenvolvido a partir da plataforma LARIISA (OLIVEIRA et al., 2010). Trata-se de um sistema inteligente de governança para o apoio à tomada de decisão em ambientes de saúde, com foco no projeto Rede Cegonha do Ministério da saúde cujo objetivo é desenvolver ações de acompanhamento de mães e da crianças, em especial nos primeiros anos de vida (LOPES; VIDAL; OLIVEIRA, 2016; SILVA et al., 2017). O AWF do GISSA é constituído por uma série de componentes que possibilitam a coleta, a integração e a visualização de informações relevantes aos procedimentos de tomada de decisão (ANDRADE; OLIVEIRA; RAMOS, 2015).

A Figura 4 exibe a tela inicial do GISSA. No menu, à esquerda, encontram-se as principais funcionalidades: Inteligência Epidemiológica, Inteligência Administrativa Financeira, Inteligência Normativa, Inteligência da Gestão do Conhecimento, Inteligência da Gestão Compartilhada/DSS e Administração.

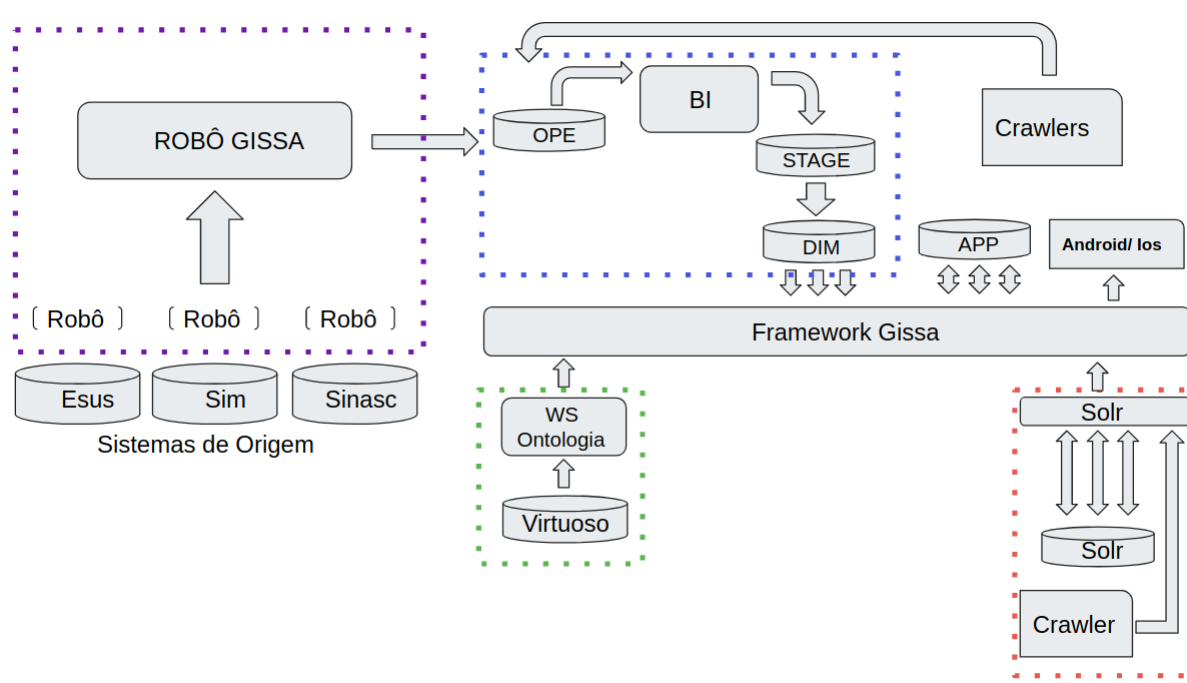
Figura 4 – Menu Inicial Gissa.



Fonte: Sistema GISSA.

A PoC do GISSA foi realizada na cidade de Tauá - Ceará, tendo havido uma expansão do sistema para 20 municípios dos estados do Ceará e Pernambuco. A Figura 5 mostra a arquitetura do GISSA, seus componentes, conexões e fluxo de dados.

Figura 5 – Arquitetura do GISSA.



Fonte: Sistema GISSA.

As informações usadas no GISSA são oriundas do Departamento de Informática do Sistema Único de Saúde (DATASUS): Estratégia de Informatização do SUS (e-SUS), Sistema de Informação de Mortalidade (SIM), Sistema de Informação de Agravos de Notificação (SINAN), Sistema de Informação do Programa Nacional de Imunização (SIPNI), Cadastro Nacional de Estabelecimento de Saúde (CNES) e Sistema Nacional de Nascidos Vivos (SINASC). Essas informações são obtidas pelo GISSA por meio de autorização junto aos responsáveis de cada município. Para o povoamento das informações do GISSA é necessária a execução da carga periódica por meio dos robôs. Esses robôs são instalados nas máquinas dos municípios que têm as bases de dados e, assim, alimentam o GISSA sazonalmente.

As informações oriundas de cada município são gravadas no banco de dados Operacional (banco de dados com informações brutas e não tratadas) do GISSA. Em seguida, estas informações são passadas para o componente de *Business Intelligence* (BI) e que usa *crawlers* (mecanismo usado para facilitar *insights* importantes para as estratégias de coleta de informações).

O componente de BI do GISSA cria as rotinas a serem refinadas nos bancos de dados *Stage* e, na sequência, no Dimensional, mantendo o registro dos dados a serem consultados no formato final e exibidos nos aplicativos do sistema. O fluxo do robô é realizado pelo *Extract, Transform and Load* (ETL) até o framework do GISSA o qual é assistido e supervisionado pelo banco de dados da aplicação APP. Este resume as

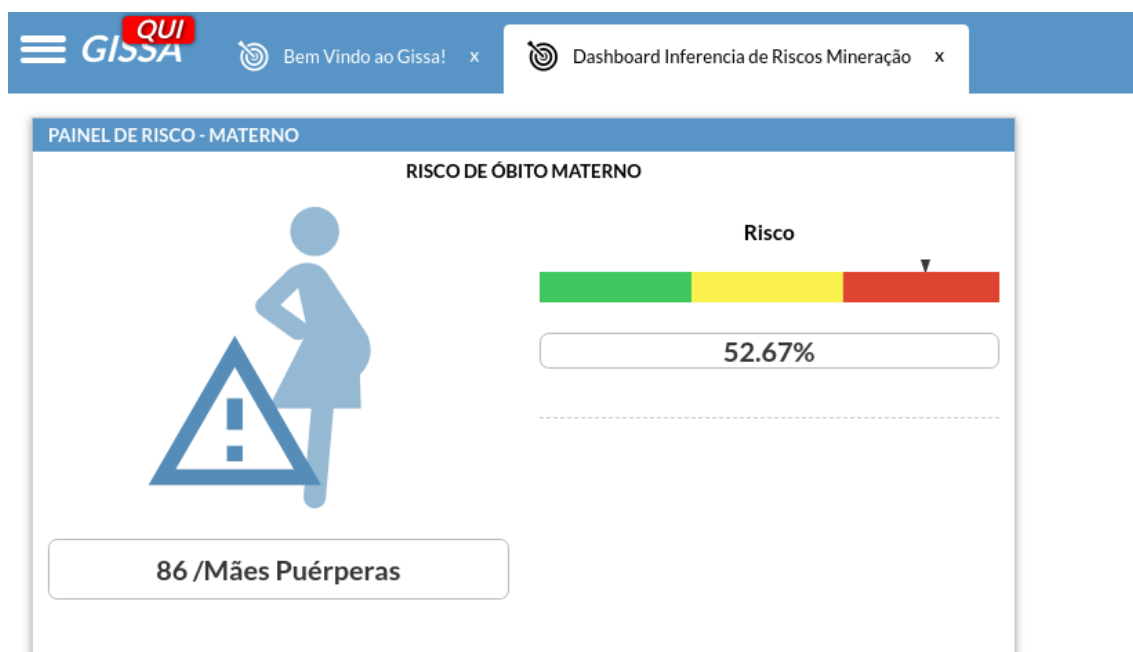
informações do sistema e trata as questões de gestão e supervisão dos dados, sendo o principal banco de dados da aplicação GISSA. Nele, ficam armazenadas informações da aplicação como usuários, níveis de acesso, municípios cadastrados e itens do menu.

O AWF é um framework proprietário usado pelo GISSA para criar seus serviços fundamentais: *dashboard*, indicadores, relatórios e alertas. Enquanto o Framework GISSA representa a aplicação web do GISSA. Pode-se dizer que todos os demais componentes existem para fornecer algum dado para a aplicação principal ou para auxiliá-la de alguma forma.

O framework do GISSA tem o potencial de agregar funcionalidades e serviços por meio da tecnologia *REST* em um estágio inicial para renderização (via *POST*). Atrelado ao AWF tem os recursos Web Semântica (WS), Solr e Android/Ios, explicados a seguir. O WS ontologia agrega valores de dados utilizando técnicas semânticas ao banco de dados Virtuoso, o qual guarda informações com objetivo de realizar a inferência de dados. O Solr realiza a indexação de determinados dados promovendo a escalabilidade dos mesmos e os armazenando no banco de dados Solr, tendo como apoio um *crawler* para capturar dados relevantes para otimização de alto volume de informações. Existem aplicativos do GISSA para Android e para IOs. Esses aplicativos exibem a página WEB do GISSA e implementam funções como roteamento de alertas.

As Figuras 6, 7, 8 e 9 mostram, respectivamente, o *dashboard* de óbito materno, relatório de óbito materno, indicadores de mortalidade geral e alertas Rede Cegonha.

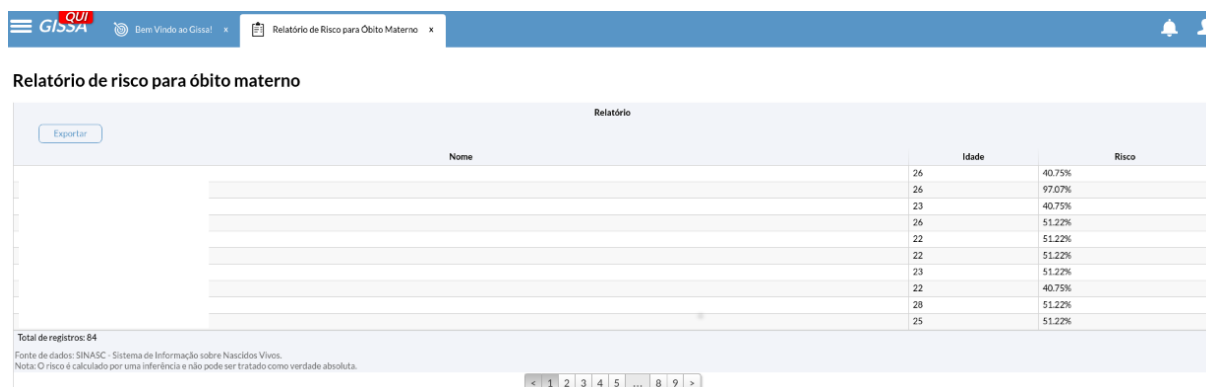
Figura 6 – Dashboard risco de óbito materno.



Fonte: Sistema GISSA.

A Figura 6 mostra a tela do *dashboard* do painel de risco materno. Neste *dashboard* há duas informações importantes que o GISSA dispõe, a quantidade (86) de mães puérperas e o risco de óbito médio das mesmas expresso em porcentagem e categorizado (baixo, médio e alto) em uma escala representada pelas cores verde, amarela e vermelha. A finalidade deste dashboard é filtrar a quantidade de mães puérperas, realizar a média aritmética dos riscos de óbitos e exibir as informações de acordo com a Figura 6.

Figura 7 – Relatório de risco para óbito materno.



Nome	Idade	Risco
	26	40.75%
	26	97.07%
	23	40.75%
	26	51.22%
	22	51.22%
	22	51.22%
	23	51.22%
	22	40.75%
	28	51.22%
	25	51.22%

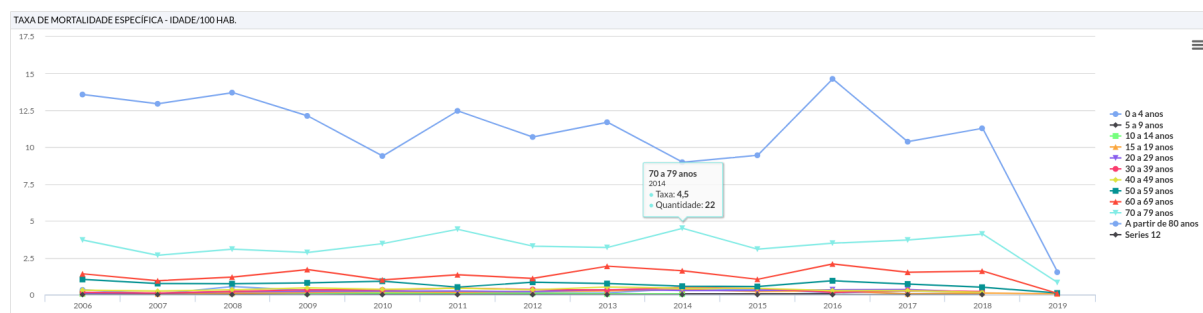
Total de registros: 84
 Fonte de dados: SINASC - Sistema de Informação sobre Nascidos Vivos.
 Nota: O risco é calculado por uma inferência e não pode ser tratado como verdade absoluta.

Fonte: Sistema GISSA.

A Figura 7 apresenta a tela de relatório de risco para óbito materno. Este relatório contém três colunas: Nome (apagado por não ser fictício), idade e risco de óbito médio da mãe.

Atualmente os relatórios do GISSA têm paginação de dez linhas e número variável de colunas, entretanto podem ser configurados para trazer mais informações. A finalidade deste relatório é exibir informações relacionadas ao risco de óbito de cada mãe, tendo, inicialmente, nome, idade, risco de óbito da mãe, mas pode-se adicionar outros dados, tais como: escolaridade, estado civil, naturalidade e etc.

Figura 8 – Indicador de taxa de mortalidade específica - idade/ 100 Hab.

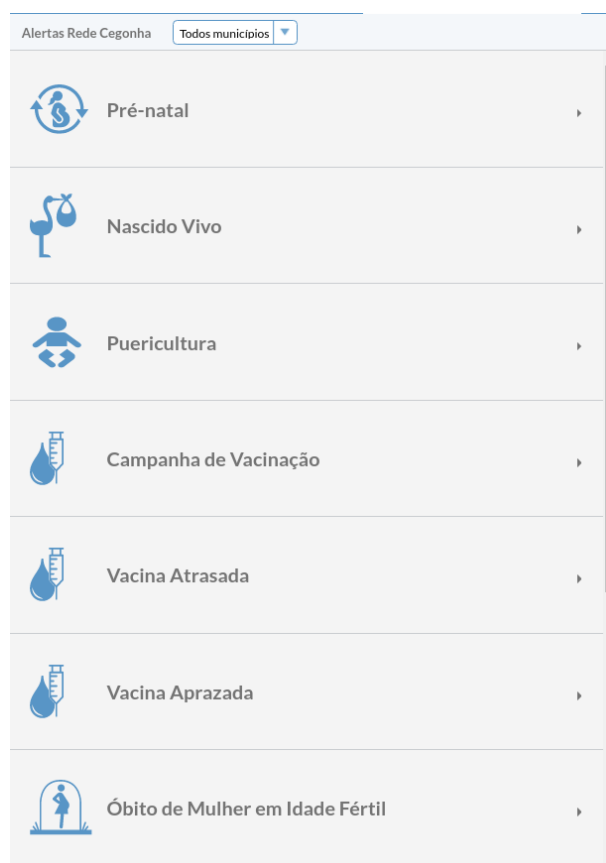


Fonte: Sistema GISSA.

A Figura 8 mostra a tela do indicador cujo nome é taxa de mortalidade específica -

idade/ 100 HAB. Seu gráfico é feito pela taxa de mortalidade (eixo Y) por ano (eixo X). No lado direito do indicador tem-se as faixas etárias (Legendas). Para cada faixa etária tem-se uma cor e um símbolo para diferenciar os gráficos. Ao clicar em um ponto específico de cada gráfico são apresentadas as informações: Faixa etária do gráfico (neste exemplo é a faixa etária 70 a 79 anos), ano (2014), taxa (4,5) e quantidade (22).

Figura 9 – Alertas disponíveis no GISSA.



Fonte: Sistema GISSA.

A Figura 9 mostra a tela dos alertas (Pré-natal, Nascido Vivo, Puericultura, Campanha de Vacinação, Vacina Atrasada, Vacina Aprazada, óbito de Mulher em Idade Fértil) disponibilizados no GISSA. Estes alertas são disparados pelo GISSA via email ou notificação para todos ou um número específico de municípios.

É importante salientar que os recursos dispostos nas Figuras 6,7 e 8 (*dashboard*, relatório e indicador) não são os únicos em operação no GISSA. Há outros tipos de *Dashboards*, relatórios e indicadores com mais informações e de forma personalizada, tendo assim um grande fluxo de informação que nem sempre são triviais de tratar e integrar.

O GISSA não é um SIS legado. Trata-se de sistema de desenvolvimento proprietário tendo o seu framework sido desenvolvido em uma tecnologia própria apresentando muitos cenários complexos e apropriados para a realização do estudo sobre integração de novos serviços.

3 Trabalhos Relacionados

Neste capítulo, foram elencados trabalhos em três categorias: (i) Disponibilização de informações via *Application Programming Interface Representational State Transfer* (*API REST*) usando a ideia de modelos; (ii) Plataforma interoperável em sistemas de informação; e (iii) Interoperabilidade em sistemas de informação na saúde.

3.1 Disponibilizando Informações via *API REST* com Modelos

Em (SELLAMI; BHIRI; DEFUDE, 2014) é proposto um modelo de recursos gerenciados pela *Unified REST API for Relational and NoSQL Data Stores* (ODBAPI). Trata-se de uma *API REST* simplificada e unificada que permite executar operações de *Create, Read, Update e Delete* (CRUD) em diferentes bancos de dados NoSQLs e relacionais, para representar os diferentes elementos de dados heterogêneos armazenados em nuvem.

Os autores (GESSERT et al., 2014) descrevem um trabalho que motiva o projeto de uma *API REST*, bem como os desafios de fornecê-la de uma maneira extensível, escalável e com alta disponibilidade. Para esse fim, é proposto a *Scalable and Unified REST API for Cloud Data Stores* (ORESTES), consistindo em um *middleware* baseado em uma camada independentemente e escalável de servidores *Hypertext Transfer Protocol* (HTTP) que mapeiam a *API REST* para armazenamentos de dados orientados a agregados (NoSQL).

O trabalho realizado por (MARINOS; WILDE; LU, 2010) apresenta uma solução que conecta o modelo de banco de dados relacional com a arquitetura web. Nela os bancos de dados relacionais são acessíveis sem etapas intermediárias e sem metadados necessários, permitindo a seus usuários executar tarefas que anteriormente exigiam uma ponte entre vários domínios e paradigmas ou que não eram possíveis.

Em (RODRÍGUEZ-ECHEVERRÍA et al., 2013) é apresentada uma abordagem orientada a modelo para criar uma *API REST* de um software web. Essa abordagem parte de um modelo conceitual do aplicativo herdado e gerado por técnicas de engenharia reversa. Neste trabalho, detalha-se o processo de geração da API e é fornecido uma implementação possibilitando uma das estruturas de desenvolvimento da web estudadas para avaliar a adequação da abordagem.

Os autores (POLÁK; HOLUBOVÁ, 2015) realizaram um trabalho sobre a geração de recursos REST com base no princípio *Model-Driven Architecture* (MDA), permitindo o projeto e a manutenção de sistemas complexos. É apresentada, também, uma abordagem para descrever a *API REST* no MDA e como fornecer gerenciamento automático da

evolução entre as versões subsequentes da API derivadas da versão original.

Diversas ferramentas comerciais, por exemplo, G9¹, OData² e SlashDB³, possuem funções semelhantes aos citados trabalhos relacionados, em que se trabalha a ideia de modelos e disponibilização de informações da base de dados via tecnologia REST com a finalidade de se tratar a acessibilidade e a automação dos dados.

3.2 Plataforma Interoperável em sistemas de informação

No trabalho (NASCIMENTO; CARVALHO; ROCHA, 2013) é proposta uma ferramenta que sistematiza uma solução de interoperabilidade entre sistemas baseados em estruturas distribuídas de modelos distintos, permitindo que os componentes envolvidos nas aplicações distribuídas conversem por intermédio de mecanismos automatizados de interoperabilidade.

Em (PERUMAL; RAMLI; LEONG, 2011) é apresentada uma plataforma de interoperabilidade para a execução de sistemas domésticos inteligentes. O núcleo da estrutura é baseado na tecnologia SOAP, fornecendo a integração independente de plataforma entre sistemas heterogêneos.

Foi realizada uma pesquisa em (AMIN et al., 2020) para a construção de uma plataforma para viabilizar a interoperabilidade de dados e a troca de informações entre os sistemas de e-saúde. A interoperabilidade é possibilitada mediante a utilização do paradigma de *Service-Oriented Architecture* (SOA).

Os autores (MIYOSHI et al., 2018) propõem a *eHealth-Interop*, uma plataforma de interoperabilidade para integração de informações de cuidados mentais. O modelo de informações de interoperabilidade usado na plataforma, baseia-se nos padrões de saúde internacional e brasileira. O ambiente de interoperabilidade proposto conseguiu integrar informações na rede de assistência à saúde mental com o apoio de cinco organizações.

Em outra plataforma interoperável, os autores (JAYARATNE et al., 2019) apresentam um ambiente para integração de dados de saúde centrado no paciente e para tomada de decisões clínicas. A plataforma foi implementada para atender os requisitos de interoperabilidade de um hospital em Melbourne, Austrália. O ambiente proposto mostrou-se promissor, pois conseguiu integrar e flexibilizar o acesso às informações dos pacientes.

Observa-se com esses trabalhos a aderência da ideia de implementar plataformas interoperáveis, justificando assim a proposta do trabalho de implementar a PIXEL, com

¹ <https://www.esito.no/en/products/g9/>

² <https://www.progress.com/blogs/restify-databases-expose-your-database-to-rest-web-services>

³ <https://www.slashdb.com/>

rápida instalação, fácil integração das informações e visando a solução de problemáticas envolvidas dos níveis das interoperabilidades.

3.3 Interoperabilidade em Sistemas de informação na Saúde

Em (FARINELLI; ALMEIDA, 2014) é descrita uma pesquisa, cuja finalidade é fazer o aprofundamento das questões identificadas e correlatas aos problemas de interoperabilidade do tipo semântica sobre o registro de informações do paciente no padrão openEHR. Os resultados expressam que o uso de ferramentas específicas pode ser relevante na busca por interoperabilidade semântica entre sistemas e a informalidade na definição dos termos que vão ser usados em vocabulários de sistemas de informação, podem impedir de ter uma compreensão clara do significado desejado.

O estudo realizado (SARIPALLE, (2019)) fornece uma visão geral do *Fast Healthcare Interoperability Resources* (FHIR) da HL7, seus conceitos, fluxo de informações e o estado da arte sobre seu status atual, utilização e adesão. Com base na pesquisa detalhada e no estado da arte, o artigo chega à conclusão de que o FHIR tem capacidade para tratar a problemática das questões de interoperabilidades entre a crescente quantidade de entidades diferentes e diversificada na área da saúde.

Em (SILSAND, 2017), a autora aborda os desafios da integração de sistemas heterogêneos de informações em saúde, com o objetivo de alcançar a interoperabilidade semântica das informações dos pacientes dentro e entre todos os hospitais. O trabalho descreve um processo complexo de desenvolvimento e integração e analisa uma estratégia promissora do uso de arquétipos do openEHR, como uma arquitetura para atingir a interoperabilidade.

A autora (BEIR, 2018) apresenta a criação de um software denominado SmartBeat com iniciativas de interoperabilidade, cuja finalidade foi o desenvolvimento e avaliação de um sistema inteligente para a gestão da Insuficiência Cardíaca em pessoas.

O trabalho feito por (LIMA et al., 2016) descreve uma arquitetura de software para desenvolvimento de um sistema EHR baseado em SOA, levando em consideração a interoperabilidade entre sistemas legados. Os resultados indicam que na literatura há uma deficiência justamente na definição de uma arquitetura interoperável para sistemas específicos.

Por meio dos trabalhos apresentados nesta seção, percebe-se a relevância de soluções de interoperabilidade envolvendo em SIS. Diferente dos trabalhos relacionados apresentados, A PIXEL apresenta uma solução voltada para a interoperabilidade na área da saúde visando alcançar mais de um nível de interoperabilidade proposto pela HIMSS e estabelecer integrações em vários contextos de interoperabilidade na área da saúde.

3.4 Quadro Comparativo

Esta seção descreve um quadro comparativo entre os trabalhos apresentados neste capítulo, relacionadas à plataforma interoperável em sistemas de informação na área da saúde. Foram apresentados trabalhos que exploraram a questão da disponibilização dos dados via *API REST* junto à ideia de modelo de comunicação.

A Tabela 1 mostra a comparação entre os trabalhos analisados e uma lista de requisitos desejáveis para a plataforma proposta no contexto de interoperabilidade em SIS. As duas primeiras colunas da tabela se referem ao estudo realizado na área de interoperabilidade. A primeira coluna sinaliza quais trabalhos tratam de algum tipo de interoperabilidades, enquanto na segunda mostra os que apresentam a sistematização de soluções por meio de arquiteturas. As três últimas colunas versam sobre plataformas propostas e suas abordagens de solução. A coluna três indica a utilização de *API REST* visando a integração entre sistemas, por suas facilidades, massiva utilização em integrações de dados, praticidade em disponibilizar informações, enquanto a coluna quatro trata a utilização da ideia de modelos, e na coluna cinco, as propostas de plataformas interoperáveis.

Três aspectos importantes são extraídos da Tabela 1: (1) As pesquisas tendem bastante a realizar a tipificação de interoperabilidade, (2) utilizam-se soluções para os problemas abordados por meio da formalização de arquiteturas e (3) aplica-se modelos para trabalhar a automatização das informações.

Percebe-se que as soluções apresentadas não atendem a todos os requisitos destacados. No entanto, todos os requisitos presentes na Tabela 1 são endereçados e atendidos pela plataforma proposta PIXEL, solução proposta e implementada nesta dissertação.

Tabela 1 – Comparação dos trabalhos relacionados com os requisitos desejados.

Trabalhos	Níveis de Interoperabilidades	Arquitetura de Solução	Utilização de <i>API REST</i>	Uso de modelos de informações	Plataforma Interoperável
Sellami, Bhiri and Defude (2014)	x	x	x	x	
Gessert et al (2014)		x	x		
Marinos et al (2010)			x	x	
Rodriguez-Echeverria (2013)	x	x	x	x	
Polák (2015)		x	x	x	
Do Nascimento et al (2013)		x	x	x	x
Perumal et al (2011)	x	x	x		x
Hidalgo Delgado et al (2019)		x	x		
Miftakul et al (2020)	x	x			x
Nodehi et al (2015)	x	x		x	x
Farinelli and Barcellos (2014)	x			x	
Saripalle and Rishi (2019)	x	x	x	x	
Silsand and Line (2017)	x	x	x	x	
Margarida (2018)	x	x	x	x	

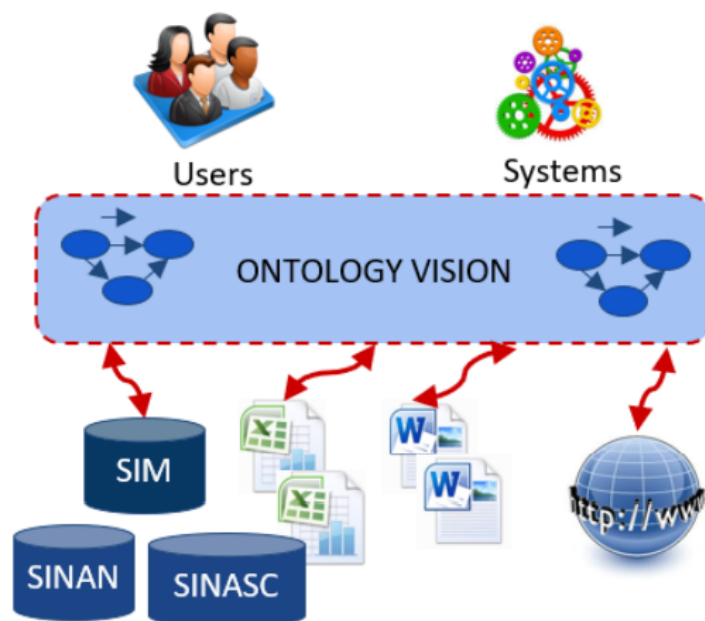
4 Estudo de Interoperabilidade do sistema GISSA

Este capítulo apresenta um estudo sobre o conceito de interoperabilidade no contexto do sistema GISSA e faz uma análise dos seguintes cenários implementados, mas ainda não integrados à sua plataforma: 1) anexação do Portal semântico baseado na utilização de ontologias, 2) incorporação de padrões de RES, 3) agregação de novos serviços de outros SIS. Diante dos aspectos de interoperabilidade e integração identificados nestes três domínios no GISSA, são analisadas as particularidades dos mesmos e, com isso, propostas soluções no contexto da plataforma PIXEL proposta.

4.1 Portal semântico no GISSA

A Figura 10 mostra o Portal Semântico desenvolvido para o sistema GISSA (CRUZ et al., 2019). Este portal fornece aos usuários e sistemas associados ao GISSA uma visão ontológica dos dados de um mundo heterogêneos existentes em banco de dados, web, etc., ou seja, fornece uma visão de um modelo de objetos única (sem dubiedade semântica) dentro de um domínio e relacionamentos num contexto homogêneo que permite a realização de inferência sobre os objetos deste domínio.

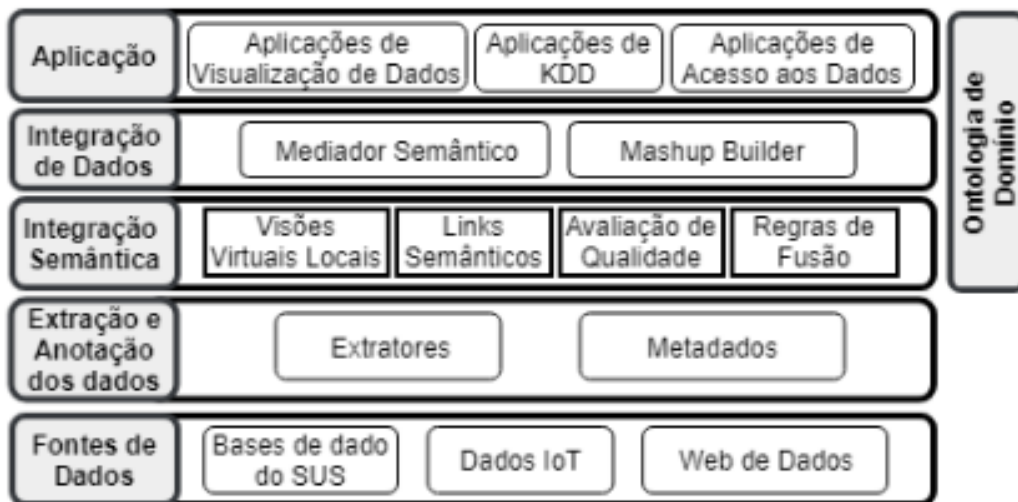
Figura 10 – Portal Semântico.



Fonte: (CRUZ et al., 2019).

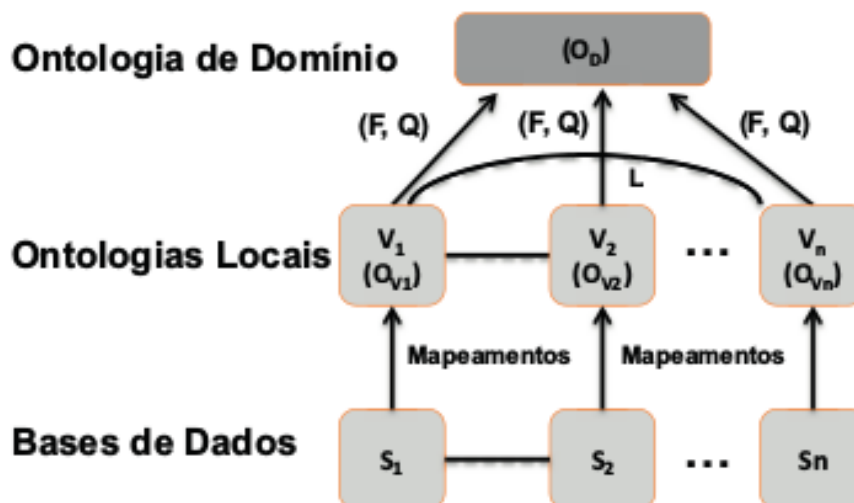
A finalidade deste portal no GISSA é, portanto, disponibilizar uma camada ontológica que liga dados semanticamente e possibilita o acesso integrado a fontes de dados diversificados (banco de dados, IoT, web etc.). O acesso a essa camada por intermédio do portal ocorre através de distintos tipos de interfaces de consulta, de tal forma que o portal possa atender diferentes solicitações de acesso e tipos de usuários. As Figuras ?? e 12 apresentam, respectivamente, a arquitetura do Portal Semântico e o framework de integração do SemanticSUS (CRUZ et al., 2019).

Figura 11 – Arquitetura do Portal Semântico.



Fonte: (CRUZ et al., 2019).

Figura 12 – *Framework* de integração semântica.



Fonte: (CRUZ et al., 2019).

A seguir, são descritas as camadas da arquitetura do Portal Semântico:

- **Fontes de dados:** são coletados dados de distintos tipos e proveniência diversificados (banco de dados de SIS diversos, dados de sensores IoT, dados da web, etc.);
- **Extração e anotação de dados:** realiza-se a extração e anotação das fontes de dados coletados na camada anterior. As anotações são metadados que delineiam desde o formato da fonte até comentários sobre os significados de elementos do esquema usado pela fonte;
- **Ontologia de Domínio:** Essa ontologia é utilizada como uma ponte para propiciar regras formais e explícitas a serem compartilhadas para a anotação semântica das fontes de dados. A linguagem utilizada para modelar a ontologia é a *Ontology Web Language* (OWL);
- **Integração Semântica (IS):** O resultado da IS define uma visão virtual e integrada das muitas fontes de dados. Na IS, usa-se o framework baseado em 3 componentes (Figura ??). O resultado da IS é uma n-tupla: $\Lambda = (O_D, V_1, \dots, V_N, L, F, Q)$, onde:
 - O_D é a ontologia de Domínio, utilizada como uma ponte para propiciar regras formais e explícitas a serem compartilhadas para a anotação semântica das fontes de dados. A linguagem utilizada para modelar a ontologia é a *Ontology Web Language* (OWL);
 - $V_1, \dots, V_N$ são especificações das visões virtuais locais descritas por ontologias locais;
 - L é uma série de regras de *linkage* (links semânticos) definidas nas classes semanticamente análogas das ontologias locais;
 - F é uma série de regras que vão especificar o processo de fusão de objetos relacionados por meio de um link OWL em uma única representação;
 - Q são métricas para avaliar a qualidade, sendo usadas para fazer a quantificação da qualidade das fontes de dados.
- **Integração de Dados:** o Portal Semântico do GISSA fornece dois serviços (Mediador Semântico e *Mashup Builder*) de acesso às fontes de dados locais por intermédio da visão ontológica:
 - **Mediador Semântico:** Os usuários definem consultas *ad-hoc* sobre a ontologia de Domínio. O mediador semântico, fazendo uso da ontologia e dos mapeamentos, reestrutura a consulta sobre as fontes de dados. Ao usuário é retornado o resultado das subconsultas consolidadas e integradas;
 - **Mashup Builder:** Os usuários especificam a visão do *Mashup* de forma especializada sobre a ontologia de domínio por meio da interface

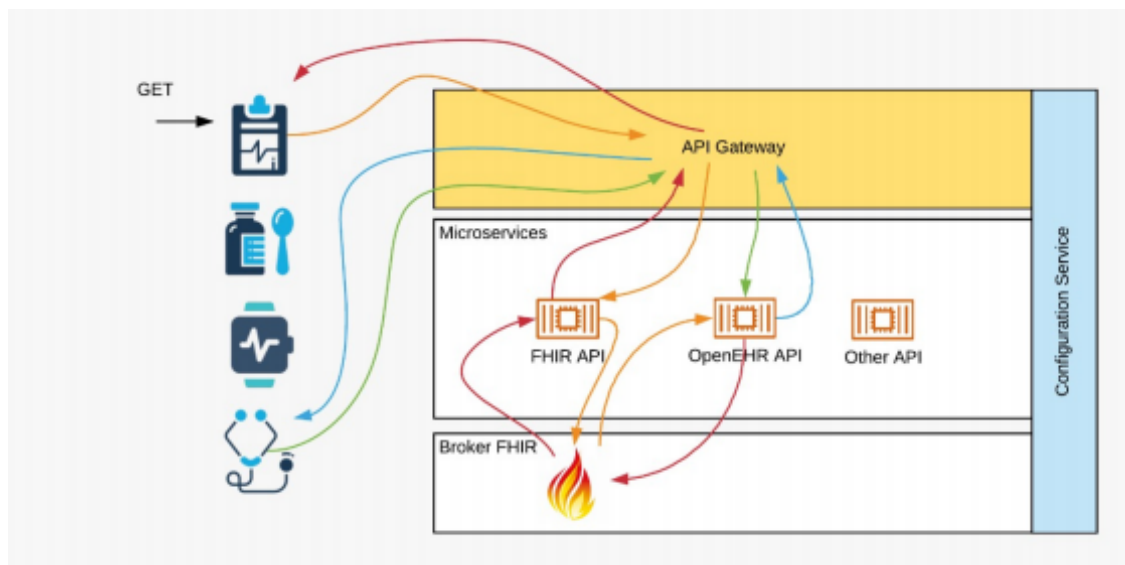
visual, e o *Mashup Builder* irá efetivar a visão de *mashup* automaticamente, de acordo com o resultado da integração semântica.

- **Aplicação:** Na última camada estão os usuários e aplicações que utilizaram os serviços de integração de dados.

4.2 RES no GISSA

A interoperabilidade semântica é a capacidade de sistemas diferentes consumirem dados com padrões divergentes, por exemplo, os padrões OpenEHR e FHIR abordados no item 3.3. Para tanto, foi construído uma espécie de *API gateway* (Figura 13) baseada em microsserviços com finalidade de disponibilizar *endpoints* para oferecer esta interoperabilidade semântica na integração entre SIS. Esta *API gateway* fará uso das camadas de microsserviços, *Broker FHIR* (GOMES et al., 2019), e de configuração de serviço. A união das funcionalidades dessas camadas para prover interoperabilidade semântica é aqui definida como um *hub* semântico.

Figura 13 – *API gateway*.



Fonte: (GOMES et al., 2019).

Assim, a proposta de criar um hub semântico capaz de fazer a integração nos múltiplos sistemas da rede de saúde privada e pública visa, principalmente, assegurar que, independente do padrão de RES em uso, as informações sejam consumidas por profissionais de saúde.

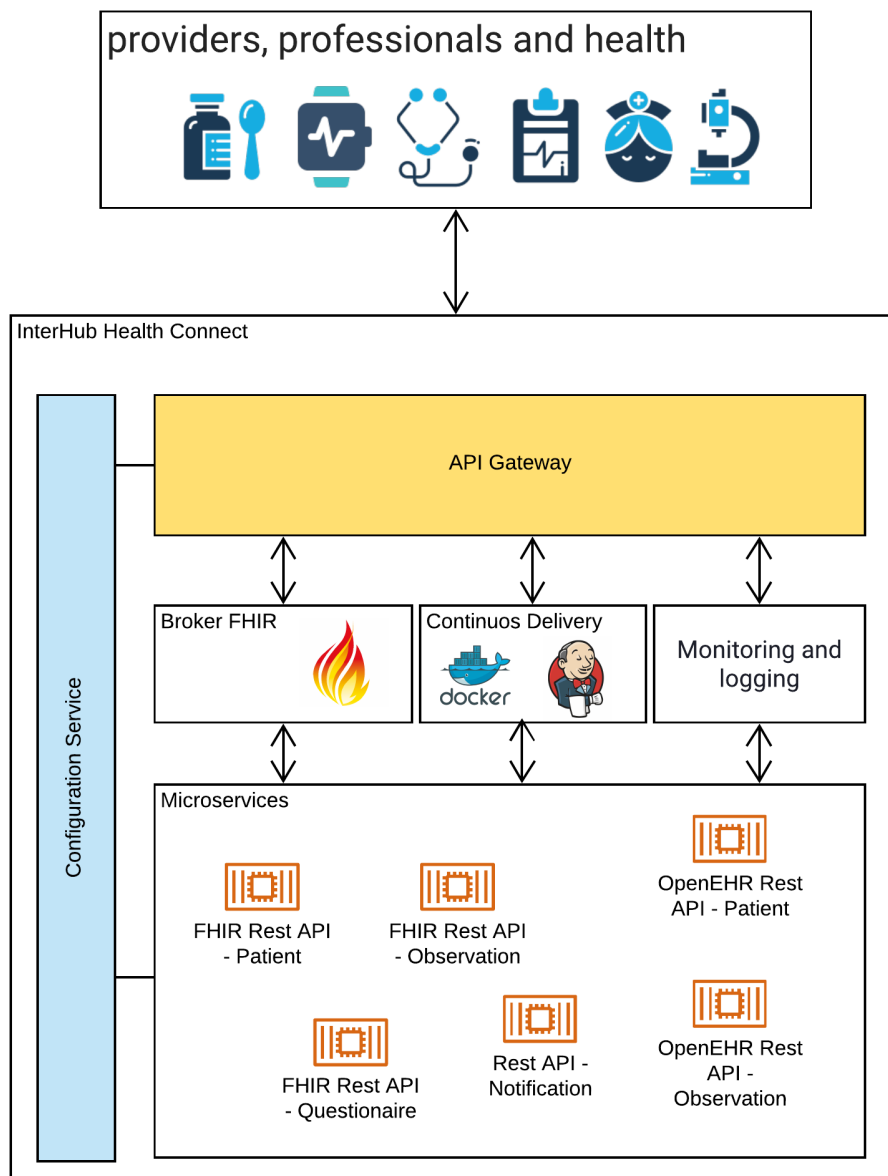
Com a arquitetura deste hub semântico nota-se que o uso de microsserviços é vantajoso nas dimensões de escalabilidade, implementação de componentes de um único sistema, uso de múltiplas linguagens, *deploys* descomplicados em componentes separados

e mais leves. Esta arquitetura usa o Modelo de Maturidade de Richardson (FOWLER, 2010), alcançando os quatro níveis de implementação de uma *API RESTful*:

- Level 0 - *The Swamp of POX*: neste nível não há necessidade de padrão; usa-se o HTTP como meio de transporte para interações remotas;
- Level 1 - *Resources*: são cruciais para fazer o controle otimizado das requisições. No lugar de utilizar apenas um endpoint, disponibiliza-se recursos individuais;
- Level 2 - *HTTP verbs*: usa os verbos HTTP para tornar as requisições mais pragmáticas. Assim, pode-se usar todos os verbos (POST, GET, PUT, PATCH, DELETE etc.);
- Level 3 - *Hypermedia Controls*: neste último nível, é possível controlar a transmissão sobre as mídias, alterando algo na estrutura interna da API. As aplicações clientes não caem, porque os recursos são disponibilizados diretamente na URI.

A Figura 14 é uma evolução da Figura 13 e apresenta, com mais detalhes, a arquitetura do *Hub* (semântico) de Integração do GISSA, constituída pelas seguintes camadas de serviços autônomos que se complementam:

- *API Gateway*: disponibiliza os *endpoints* para troca de dados entre diversos sistemas das redes privada e pública, assim como a gerência dos microsserviços;
- *Configuration Service*: camada responsável por otimizar a comunicação e gestão dos microsserviços. Considerando-se a qualidade dos serviços faz-se necessária a criação de um serviço de configuração. Tal recurso é essencial para gerenciar dezenas de microsserviços;
- *Broker FHIR*: trata os dados de sistemas proprietários e dos padrões diferentes com o FHIR, onde realiza-se todo o mapeamento dos padrões;
- *Continuous Delivery*: Para facilitar o *deploy* e assegurar entregas contínuas de muitos microsserviços, é necessário criar *containers* dos serviços. As ferramentas Docker e Jenkins garantem uma excelente estratégia de *Continuous Delivery*;
- *Monitoring and logging*: ao utilizar a arquitetura de microsserviços, desafios como o monitoramento e o *logging* virão de componentes muitas vezes implementados com linguagens diferentes. Tal fato torna fundamental o uso de soluções capazes de monitorar/gerar log para rastrear o fluxo de dados dos serviços. Como solução, a proposta é usar o Zipkin, uma ferramenta para rastreamento distribuído, eficaz em arquiteturas de microsserviços. O Uso das ferramentas e camadas podem ser vista na Figura 14:

Figura 14 – Arquitetura do *hub* de Integração.

Fonte: (GOMES et al., 2019).

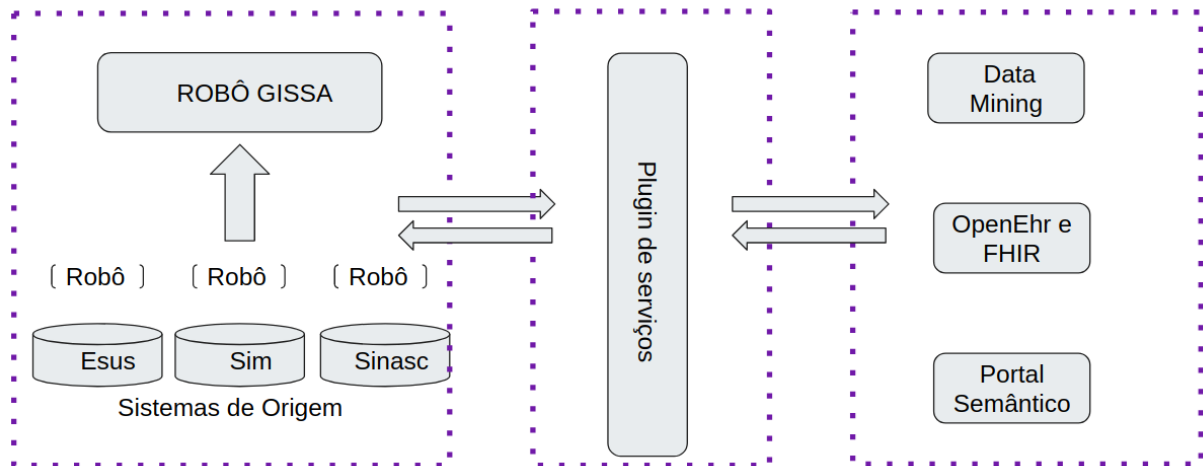
4.3 Agregação de novos serviços no GISSA

A Figura 15 mostra a utilização do barramento *plugin* de serviços da arquitetura GISSA (Figura 15) que possibilita a integração das três novas funcionalidades abordadas neste capítulo: Portal Semântico, Padrões RES e Agregação de Novos Serviços (Data Mining, no caso).

A integração foi feita a partir da tecnologia *RESTful*, via requisições *GET*, *POST* e *PUT*, que, por sua vez, foram injetadas no componente de abstração (*plugin* de serviços) mostrado no centro da Figura 15. Este barramento *plugin* de serviços é um componente de

abstração que conecta as aplicações a base de dados, permitindo a anexação de aplicações dos mais diversos tipos de tecnologias e linguagens de programação. Ele é, portanto, responsável por realizar as integrações das aplicações ao sistema, definir verbos HTTP utilizados pela tecnologia *RESTful* e a estruturação dos *endpoints* para formalizar as rotas das requisições e o consumo/disponibilização das informações.

Figura 15 – Utilização do barramento *plugin* de serviços.



Fonte: Próprio Autor.

Para a utilização do barramento de serviços, fazem-se necessárias as etapas para a realização da integração, como mostrado na Figura 16.

Figura 16 – Etapas para a integração na interoperabilidade de agregação de novos serviços.



Fonte: Próprio Autor.

- **Definir formato das Informações:** nesta etapa são trabalhadas a integração, bem como o tipo de compartilhamento de dados, os verbos HTTP e suas conexões;
- **Conduzir as Informações:** nela são trabalhados os parâmetros que serão utilizados na integração, bem como o modelo de dados que serão utilizados. Nesse momento acontecerá a condução das informações entre os sistemas envolvidos;
- **Estabelecer as conexões dos *EndPoints*:** mediante as informações já definidas e sendo possível a condução delas, faz-se necessário assegurar que as

informações sejam reconhecidas entre as pontas dos sistemas envolvidos e, desta forma, estabelecer os *endpoints*, assegurando que os sistemas vão se comunicar.

- **Exibir as Informações:** tem-se, agora, as informações definidas, conduzidas e a segurança de que elas serão usadas pelos sistemas que desejarem ter acesso. Estas informações serão preparadas e, em seguida, exibidas, possibilitando que sua integração seja visualizada.

Em conclusão, diante dos cenários de integração e interoperabilidade identificadas no GISSA, é proposta a PIXEL, Plataforma para Integração de Experimentos de Interoperabilidade em Sistemas Legados de Saúde Pública, detalhada no próximo capítulo. Trata-se, portanto de uma ferramenta de integração que provê interoperabilidade fundamental, sintática e semântica em SIS, utilizando o GISSA em sua PoC. Arquiteturalmente, a PIXEL é um conjunto de *endpoints* e uma série de configurações a serem feitas, com finalidade de disponibilizar e acessar inicialmente as informações de sistemas públicos legados e sistemas proprietários que não têm comunicação externa ao seu escopo. uma ferramenta de integração que provê interoperabilidade fundamental, sintática e semântica em SIS, tendo o GISSA como uma PoC.

5 Plataforma PIXEL

Este capítulo apresenta detalhes de concepção e implementação da PIXEL, uma plataforma para interoperabilidade em SIS legados e de implementação proprietária nos níveis fundamental, sintática e semântica. A plataforma PIXEL propõe a solução destes três aspectos de interoperabilidade citados, tendo sido implementados os aspectos de Fundamental (Consumo e Disponibilidade de Informação), Sintaxe (estrutura de Banco de Dados) e parte do aspecto Semântica (RES para OpenEHR e FHIR).

São destacados nesta seção: cenário de aplicação, requisitos funcionais e não funcionais, arquitetura, tratamento de interoperabilidade e aspectos de implementação.

5.1 Cenário de Aplicação

Embora a PIXEL possa ser usada em qualquer sistema legado ou de implementação proprietária, por questões didáticas optou-se, neste capítulo, pelo uso do GISSA como cenário de aplicação. Considere-se, portanto, o problema de interoperabilidade do sistema GISSA descrito no item 1.2 (Descrição do Problema) onde se deseja integrar um novo serviço externo.

Via de regra, todos os dados manipulados pela AWF, plataforma operacional do GISSA, são provenientes da captura de informações de banco de dados governamentais (e-SUS, SIM, SINASC, etc.), via robôs. No item 1.2 foram identificados os seguintes aspectos de interoperabilidade:

Básica: Quanto ao aspecto de interoperabilidade comunicação, são identificados dois problemas a serem considerados neste cenário de aplicação:

- **Consumo de Informação:** a comunicação do GISSA com qualquer sistema externo só pode ser feita via o *POST* (consumindo informações), único verbo disponibilizado pelo *back-end* do GISSA e, portanto, usado pelo AWF.
- **Disponibilização de Informação:** o GISSA não possui um mecanismo que disponibilize informações para uma comunicação com outro sistema ou componentes em mundo exterior ao seu.

Sintaxe: Uma vez resolvida a questão do consumo e disponibilidade da informação, o próximo passo é o tratamento sintático desta informação. No caso, trata-se do mapeamento de informações (tamanho, tipos, nomes de campos etc.) contidas em uma base de dados, transformando-as em um modelo de dados e com a possibilidade de transformação/inserção em outra base

Para prover a interoperabilidade sintática em sistemas legados de saúde pública, a PIXEL faz uso do gerador de modelos de informações, estabelecendo o mapeamento personalizado a partir das informações de uma base de dados e tabela específicas, selecionadas via interface Web da plataforma.

Semântica: Para prover a interoperabilidade semântica em SIS que usam padrões de consensos como FHIR e OpenEHR, a PIXEL usa o *Gateway* para Interoperabilidade de RES em Sistema de Baixo Custo (GIRLS) (GOMES et al., 2019).

5.2 Requisitos funcionais e requisitos não-funcionais

Os requisitos da plataforma PIXEL foram delineados a partir de *brainstormings*, reuniões e observações envolvendo participantes da equipe GISSA, realizadas durante a concepção da plataforma. O artefato concebido nesta fase foi o documento *Software Requirements Specification* (SRS) (SOARES; MOURA, 2015).

Ao se elencar os requisitos para a implementação da PIXEL, houve uma preocupação em priorizar aqueles que tratassem de perto as questões de interoperabilidades entre os distintos sistemas que desejam se comunicar, possibilitando a validação da arquitetura e da plataforma proposta. Tendo o documento de requisitos definido, foi possível elaborar um diagrama de casos de uso a partir da análise feita. Para o diagrama de casos de uso foram definidos os atores: Administrador da Plataforma e o Administrador de Sistemas.

O Administrador da Plataforma representa o ator interagindo com os sistemas que desejam compartilhar informações entre si. Para cada requisito funcional (Tabela 2) e requisito não funcional (Tabela 3), um ou mais casos de usos foram definidos, conforme pode ser visto na Figura 17.

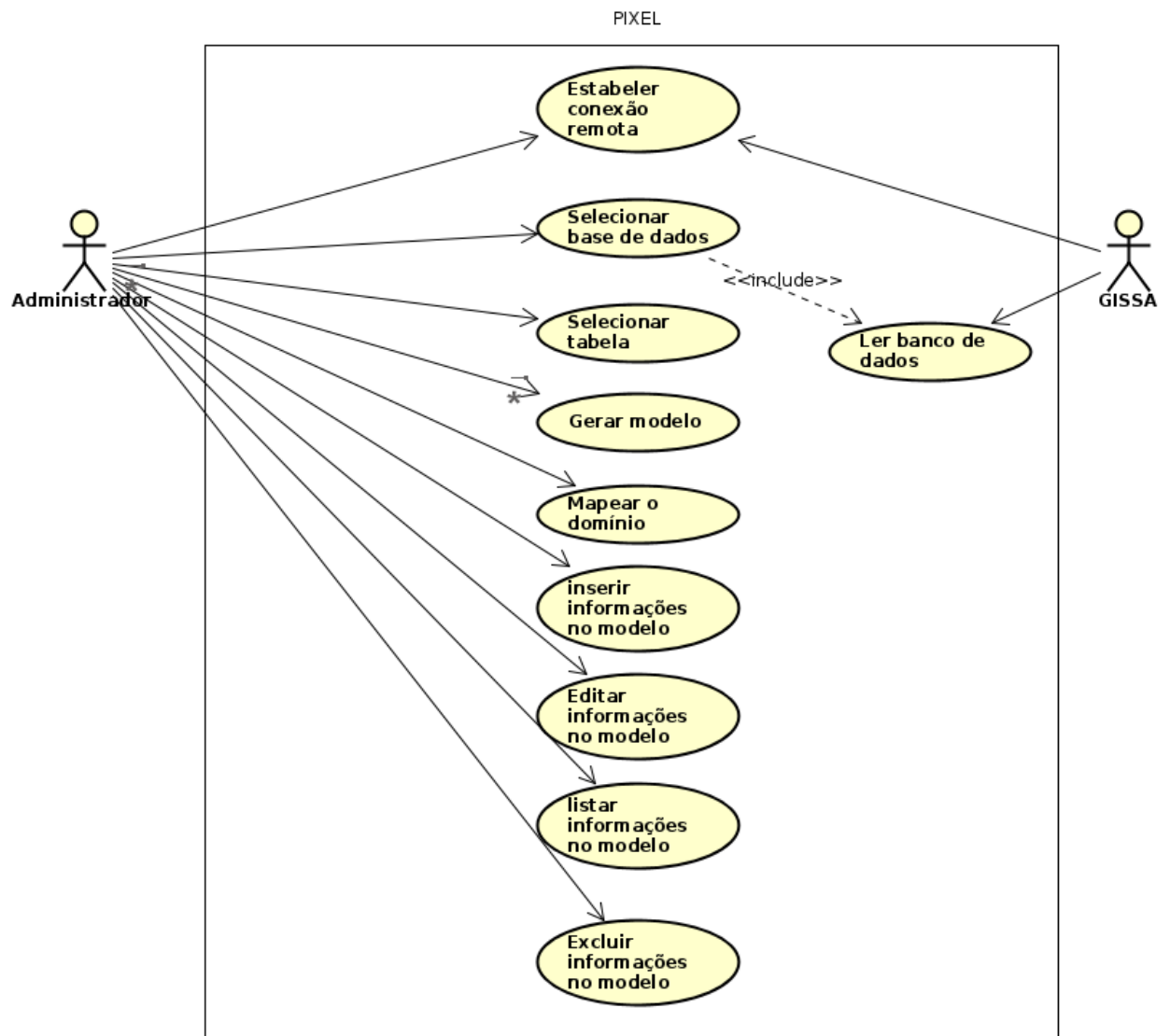
Tabela 2 – Requisitos Funcionais da PIXEL

ID	REQUISITOS FUNCIONAIS
RF1	A plataforma deverá estabelecer a conexão com a base de dados locais.
RF2	A plataforma deverá estabelecer a conexão com a base de dados remotas.
RF3	A plataforma deverá selecionar abase de dados.
RF4	A plataforma deverá selecionar tabela.
RF5	A plataforma deverá mapear o domínio.
RF6	A plataforma deverá gerar modelo.
RF7	A plataforma deverá inserir informações no modelo.
RF8	A plataforma deverá editar informações no modelo.
RF9	A plataforma deverá excluir informações no modelo.
RF10	A plataforma deverá listar informações do modelo.

Tabela 3 – Requisitos Não Funcionais da PIXEL

ID	REQUISITOS NÃO FUNCIONAIS
RNF1	A plataforma deverá funcionar em versões modernas dos principais navegadores Google Chrome, MozillaFirefox, Internet Explorer e Safari.
RNF2	plataforma deverá oferecer uma interface gráfica com foco na experiência do usuário com recursos de seleção e botões que facilite a usabilidade.
RNF3	A plataforma deverá manter o sigilo e a segurança das informações enviadas/recebidas usando a tecnologia <i>REST</i> .
RNF4	A plataforma deverá funcionar em ambientes windows e Linux.
RNF5	A plataforma deverá ser desenvolvido baseado na arquitetura cliente-servidor.

Figura 17 – Escopo de funcionamento da PIXEL.



Fonte: Próprio Autor.

Após a definição e finalização dos casos de uso, a próxima seção apresenta a arquitetura da PIXEL onde serão detalhados seus componentes funcionais.

5.3 Arquitetura PIXEL

Considere-se, para melhor entendimento da arquitetura, um cenário simples em que a plataforma PIXEL está sendo utilizada para prover interoperabilidade entre dois sistemas, A e B, desenvolvidos em tecnologias proprietárias. Neste cenário, objetiva-se mostrar no relatório do sistema A, as informações coletadas de um banco de dados do sistema B. Para tanto, serão realizados os seguintes passos:

Configuração: A iniciação da plataforma PIXEL requer as seguintes ações:

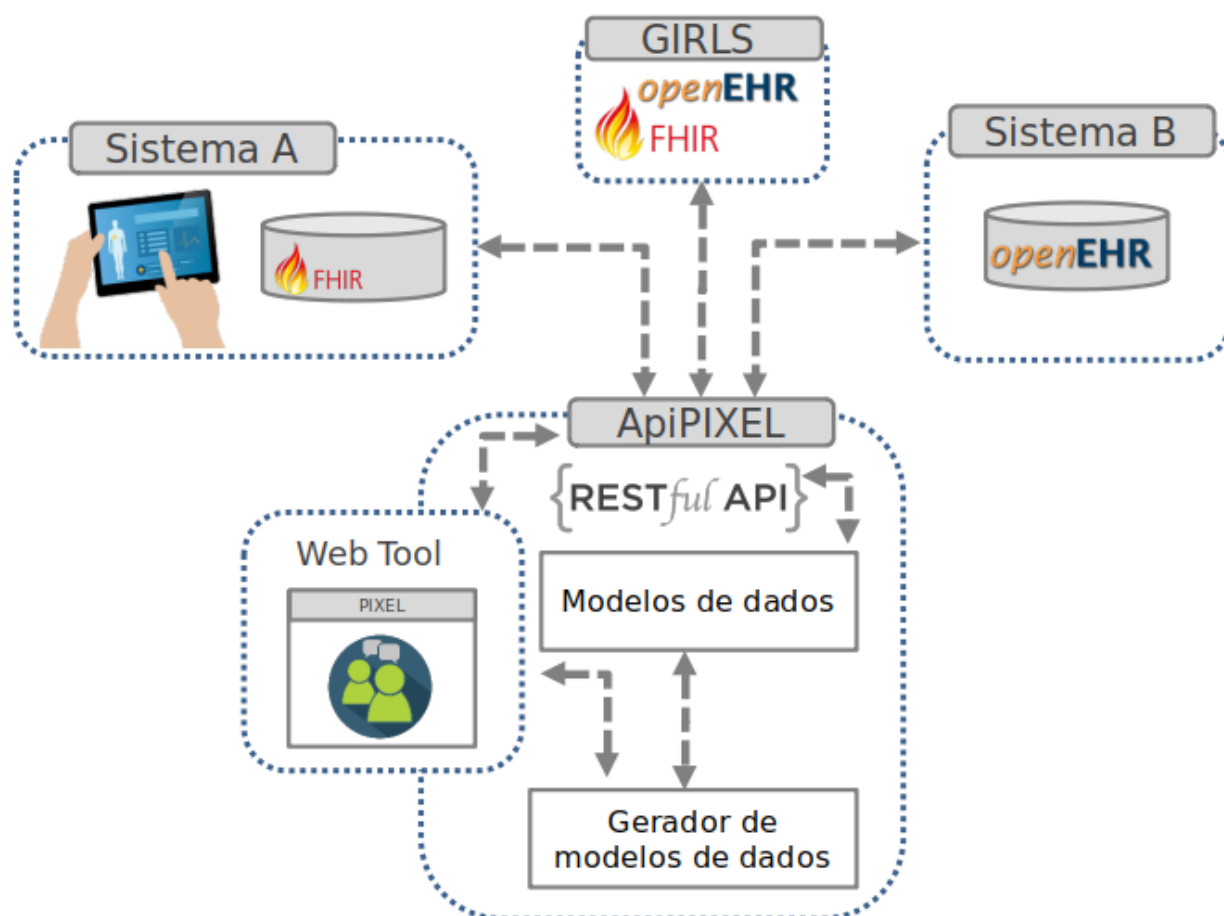
1. Cadastrar os parâmetros de conexão para o servidor desejado do sistema B;
2. Estabelecer conexão com o servidor selecionado do sistema B;
3. Listar *databases* disponíveis do servidor selecionado do sistema B;
4. Selecionar *database* desejado do sistema B;
5. Listar tabelas disponíveis da *database* selecionado do sistema B;
6. Selecionar tabela do sistema B;
7. Gerar modelo de dados a partir dos passos anteriores e tendo como parâmetro as informações da tabela selecionada do sistema B;
8. Gerar o JAR.

Execução: Uma vez configurado, o JAR é gerado e publicado, permitindo que o sistema A possua uma *API Restful* intermediada pela PIXEL.

Operação: Esta etapa ficará a cargo do sistema A, consumindo as informações e interagindo com o sistema B.

A Figura 18 apresenta os principais componentes da arquitetura da plataforma PIXEL que objetiva promover a interoperabilidades entre SIS e integrar novos serviços em sistemas legados

Figura 18 – Arquitetura da PIXEL.



Fonte: Próprio Autor.

Sistema A e Sistema B: Correspondem aos sistemas que vão conversar via PIXEL, sendo o Sistema A quem recebe as informações e o Sistemas B quem as envia.

GIRLS: *Gateway* usado para a conversão entre os padrões OpenEHR e FHIR, funcionando com uma estrutura da PIXEL para padronização das informações.

Web Tool: Interface gráfica da PIXEL onde é feita a escolha da base de dados e das tabelas usadas. As informações das tabelas são enviadas para o Gerador de Modelos.

Gerador de Modelos: mecanismo ou estrutura da PIXEL que espelha a(s) tabela(s) selecionada(s) do sistema B, realizando o(s) seu(s) mapeamento(s) e interligando os campos das informações desta(s) tabela(s) fornecendo as informações para o sistema A, possibilitando o consumo das informações da(s) tabela(s), com os campos já predefinidos e prontos para a utilização gerando o modelos de dados.

A Figura 19 apresenta o diagrama de classe que constitui o gerador de modelos. Estas classes interagem via Model de cada *Controller* de acordo com o funcionamento do padrão de projeto *Model, View and Controller* (MVC).

Figura 19 – Diagrama de classe da estrutura Gerador de Modelos.



Fonte: Próprio Autor.

A classe *BaseController* implementa as funcionalidades de selecionar, listar e especificar as bases de dados de onde serão escolhidas as tabelas para compartilhamento de informações. A classe *GerarArquivoController* é responsável por fazer o mapeamento das informações que se desejam disponibilizar/consumir, gerando *endpoints* em tempo de execução a partir da tabela escolhida.

As Figuras 20 e 21 mostram, respectivamente, os trechos de códigos das funcionalidades de listar e selecionar da base de dados.

Figura 20 – Trecho do código usado para listar a base de dados.

```

12   def listar = {
13     def objJson = Base.list() ?: []
14     def arrObjJson=[]
15     if(params.id){
16       if(Base.findById(params.id)){
17         arrObjJson.add(Base.findById(params.id))
18         render arrObjJson as JSON
19       }
20       if(!Base.findById(params.id)){
21         def json = '{"id":'+ "${params.id}" +',
22           "msg": "Base nao encontrado!"}'
23         render json
24       }
25     }
26     else{render objJson as JSON}
27   }

```

Fonte: Próprio Autor.

Figura 21 – Trecho do código usado especificar a base de dados.

```

116   def save = {
117     def base = new Base(params)
118     if(!base.hasErrors() && base.save()) {
119       flash.message = "Base ${base.id} cadastrado"
120       redirect(action:show,id:base.id)
121     }
122     else {
123       render(view:'create',model:[base:base])
124     }
125   }

```

Fonte: Próprio Autor.

A figura 22 mostra o trecho de código da funcionalidade de listar tabelas. Por fim é gerado um arquivo no pacote *Model* com todas as especificações da tabela selecionada. Com a criação desse arquivo, tem-se seus *endpoints* a partir do mesmo, com isso é criado o modelo.

Figura 22 – Trecho do código usado para listar as tabelas disponíveis.

```

18     def listar = {
19         def objJson = GerarArquivo.list() ?: []
20         def arrObjJson=[]
21         if(params.id){
22             if(GerarArquivo.findById(params.id)){
23                 arrObjJson.add(GerarArquivo.findById(params.id))
24                 render arrObjJson as JSON
25             }
26             if(!GerarArquivo.findById(params.id)){
27                 def json='{"id":'+params.id+', "msg":
28                     "GerarArquivo nao encontrado!"}'
29                 render json
30             }
31         }
32     }
33 }

```

Fonte: Próprio Autor.

Modelo de dados: Resultado do mapeamento da entidade do sistema B com suas características, propriedades e relacionamentos, possibilitando a operacionalidade no sistema A através de *endpoints RestFul* viabilizados pela PIXEL.

A Figura 23 apresenta o diagrama de classes que compõe os Modelos criado via PIXEL. Estas classes interagem via cardinalidades e relacionamentos definidos no *Model* de cada uma.

Figura 23 – Diagrama de classe da estrutura Modelos.



Fonte: Próprio Autor.

A classe *BaseModel* define o nome, descrição e o relacionamento *BelongsTo* usando o objeto de *GerarArquivo* para especificar a base que deseja utilizar. A classe *GerarArquivoModel* define o nome, tipo e o caminho do arquivo, bem como o relacionamento *hasMany* usando o objeto *Base*.

As Figuras 24 e 25 mostram respectivamente os trechos de códigos do *Model* das classes *BaseModel* e *GerarArquivoModel*.

Figura 24 – Código da classe BaseModel.

```

1  class Base {
2
3  String nome
4  String descricao
5
6  static belongsTo = GerarArquivo
7
8  String toString() {
9      |         "${this.nome}"
10     |     }
11 }

```

Fonte: Próprio Autor.

Figura 25 – Código da classe GerarArquivoModel.

```

1  class GerarArquivo {
2  String nomeArquivo
3  String tipoArquivo
4  String caminhoArquivo
5
6  Base base
7  |     static hasMany = [base:Base]
8  }

```

Fonte: Próprio Autor.

ApiPIXEL: É uma *API RestFul* orientada a modelos de dados que disponibiliza serviços, comportando-se como um *gateway* entre os dois sistemas comunicantes. As quatro principais funcionalidades da *API PIXEL* são: (1) Importar o novo modelo criado pelo Gerador de Modelo; (2) Solicitar a *API RESTful* para receber dados do sistema B usando o verbo apropriado; (3) Preencher o novo(s) modelo(s), disponível no sistema A, pelo Gerador de Modelos; (4) Solicitar a *API RestFul* para enviar dados(recebidos do sistema B), já no novo modelo, para o Sistema de Consumo de Dados do sistema A, usando o verbo apropriado. A Figura 26 apresenta o código da *API PIXEL* e seu funcionamento dinâmico, permitindo que seus *endpoints* sejam feitos em tempo de execução.

Figura 26 – Código da classe UrlMappings.

```

3  class UrlMappings {
4
5      static mappings = {
6          delete "/$controller/$id(.$format)?"(action:"delete")
7          post  "/$controller(.$format)?"(action:"index")
8          get   "/$controller/$id(.$format)?"(action:"show")
9          post  "/api/$controller(.$format)?"(action:"save")
10         put   "/$controller/$id(.$format)?"(action:"update")
11         patch "/$controller/$id(.$format)?"(action:"patch")
12
13         "/"(controller: 'application', action:'index')
14         "500"(view: '/error')
15         "404"(view: '/notFound')
16     }
17 }

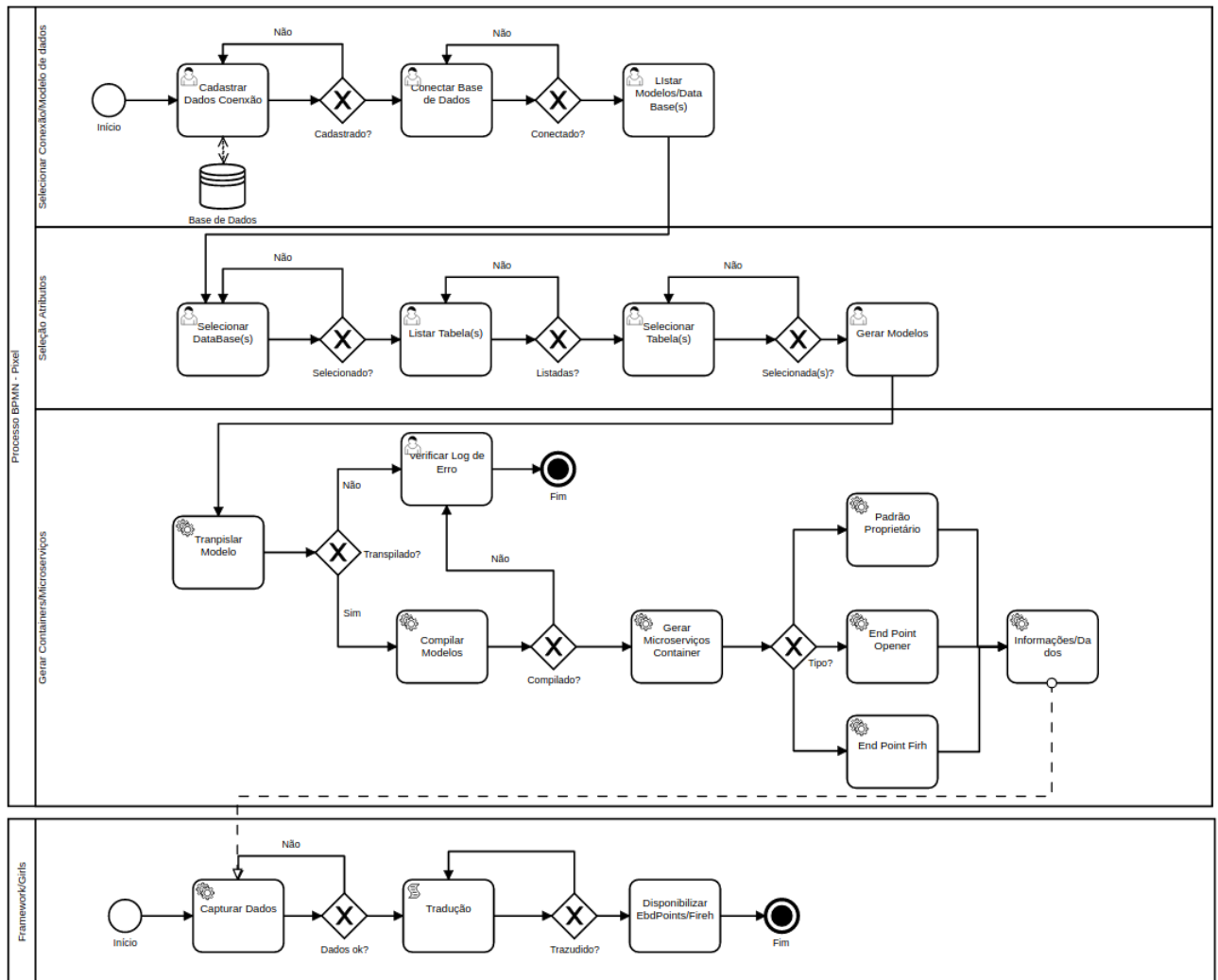
```

Fonte: Próprio Autor.

5.4 Fluxo de Funcionamento

A Figura 27 apresenta o *Business Process Model and Notation (BPMN)* do fluxo de funcionamento da PIXEL.

Figura 27 – Fluxo de funcionamento da arquitetura PIXEL em BPMN.



Fonte: Próprio Autor.

1. **Seleção de conexão:** seleciona-se a base de dados e a tabela;
2. **Seleção de atributos:** filtram-se as informações a serem convertidas em modelos de dados;
3. **Geração de serviços:** disponibiliza-se os serviços baseados (*endpoints*) nos modelos de dados escolhidos;
4. **GIRLS:** informações são enviadas para GIRLS, onde é feito o mapeamento entre os padrões OpenEHR e FHIR e retornadas as informações transcritas, seja no padrão OpenEHR ou FHIR.

5.5 Tratamento da Interoperabilidade

Considerando-se o cenário de aplicação da seção 5.1 e os componentes da arquitetura da PIXEL, é possível tratar os níveis de interoperabilidades (Fundamental, Sintático e Semântico) nos seguintes componentes:

Gerador de modelos: neste componente é tratada a interoperabilidade sintática, pois há nele as definições de nome, tamanho e tipo para gerar o modelo.

ApiPIXEL: Nesse componente é tratada a interoperabilidade fundamental, permitindo que a PIXEL intermedie e faça a comunicação dos sistemas envolvidos.

GIRLS: neste componente é tratada a interoperabilidade semântica, usando o *Broken* para conversões de padrões (OpenEHR e FHIR) e assim disponibiliza as informações convertidas via *API RESTful*.

5.6 Aspectos de implementação da PIXEL

Nesta seção são detalhados os aspectos tecnológicos usados na PIXEL. Esta seção está dividida em funcionamento tecnológico e tecnologias utilizadas na implementação da PIXEL.

5.6.1 Funcionamento tecnológico da PIXEL

A plataforma foi desenvolvida utilizando a arquitetura apresentada na seção 5.3 deste trabalho. Para implementação da proposta optou-se pela plataforma Java *Enterprise Edition* (EE) com foco em recursos da Web. A seguir, detalhamos as fases do funcionamento da PIXEL:

Comunicação: Para a efetivar a comunicação da plataforma PIXEL à base de dados dos sistemas legados, foram usadas as tecnologias *Java Database Connectivity 3* (JDBC), *pgAdmin3* e *PostgreSQL*. A comunicação se dá por meio da classe *geracaoBase.groovy*, responsável por comunicar a base de dados e suas tabelas permitindo assim a transpilação.

Geração de scaffold: Para a criação dos arquivos de configuração, diretórios, classes, entidade e pacotes da plataforma usou-se o *Gradle*. Essa opção se justifica por tratar-se de um sistema robusto de automação de compilação *open source* que facilita o trabalho ao propor uma *Domain-Specific Language* (DSL) baseada em *Groovy* em vez do XML usado pelo *Apache Maven*.

Geração de modelo: A plataforma gera o modelo por meio da classe *geracaoDominio.groovy*. Neste momento há uma correspondência direta entre as tabelas de dados

e os seus domínios. Desta forma, a relação entre as tabelas onde originam os dados e o objeto que os disponibiliza é configurada pelo usuário.

Build: Compila os recursos (bibliotecas) e o código-fonte da *API* para o empacotamento do .jar/.war. A classe responsável por realizar o *build/gradle* na plataforma chama-se *buildRest.groovy*.

5.6.2 Tecnologias e Modelagem da PIXEL

Esta seção descreve os aspectos de implementação da plataforma PIXEL. Para sua implementação, foi utilizada a plataforma Java EE com foco em recursos da Web. Foram utilizadas as seguintes especificações tecnológicas:

- *Grails 3.3.2*;
- *Servlet 2.4*;
- *Java Server Pages 2.2*;
- *Materialize 0.100.0*;
- *Java Persistence API (JPA) 2.2*;
- *Rest API 1.4*;
- *Postgresql 9.0-802 jdbc3*;
- *Postgresql 42.2.5 jre6*;
- *Jquery 3.2.1*;
- *Gradle 3.1*;
- *Java Development Kit (JDK) 1.7.0.02*;
- *Linux e Windows*;
- *Groovy 2.5.0*;
- *JSTL 1.1*;
- *pgAdmin 3*;
- *Eclipse Integrated Development Environment (IDE) 4.11*.

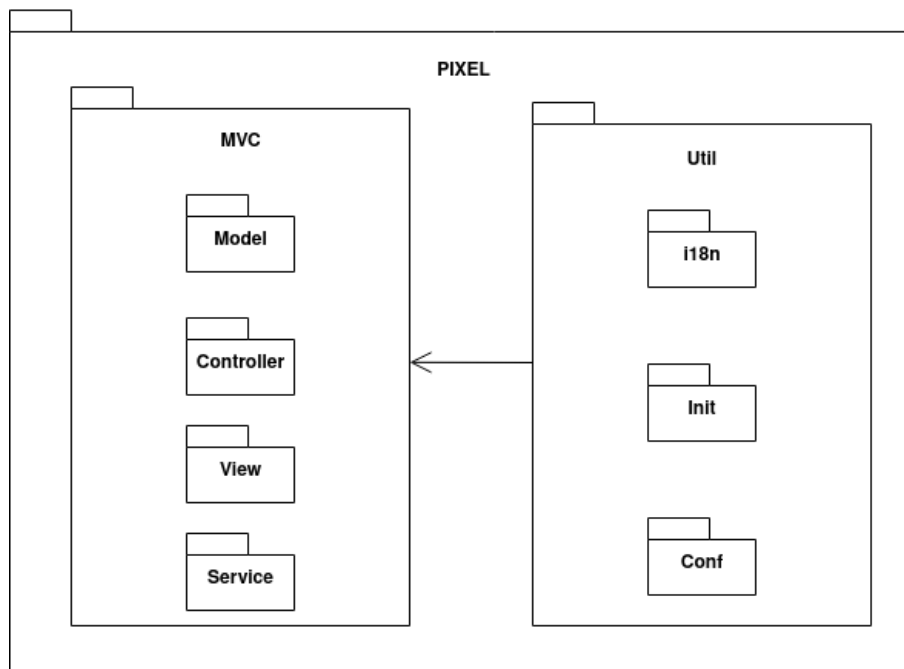
A plataforma Java EE foi escolhida porque a linguagem de programação Java tem uma grande comunidade e muitos fóruns de discussões, documentação pormenorizada, sendo uma das linguagens mais usadas e com alto grau de completude tecnológica. Em conjunto com Grails, *JavaServer Pages Standard Tag Library* (JSTL), Groovy, Gradle tem um amplo gerenciamento para *APIs*. Optou-se em utilizar o *framework JavaServer Pages* (JSP) porque possui alta abstração visual para a interface.

As demais especificações tecnológicas buscam proporcionar: Gerenciamento (Grails, Groovy e Gradle), produtividade (JSP, JSTL, Jquery, Servlet, materialize e JPA), base de dados (pgAdmin, Postgresql 42.2.5 jre6 e Postgresql 9.0-802 jdbc3), segurança na condução das informações (*API REST*) e portabilidade (Linux e Windows). Além das especificações já citadas, o desenvolvimento desta plataforma utilizou a *IDE Eclipse* e as ferramentas de

controle de versão GIT e Github.

Depois da montagem do ambiente de desenvolvimento, o passo seguinte é mostrar detalhes da implementação da plataforma PIXEL, portanto, o diagrama de pacotes da PIXEL é mostrado na Figura 28 e os diagramas de classes das estruturas MVC e Init, respectivamente, são apresentados nas Figuras 29 e 30.

Figura 28 – Diagrama de pacotes da PIXEL.



Fonte: Próprio Autor.

O diagrama de pacote da Figura 28 contém os dois pacotes que estruturam a plataforma: *MVC* e *UTIL*. O pacote *mvc* está subdividido em três pacotes:

Pacote Model: Neste pacote estão localizadas as classes que representam os modelos de dados (entidades) do sistema. Essa ação é possível somente se o usuário selecionar as entidades, ocasionando o mapeamento via PIXEL, então será gerado um modelo de dados corresponde. Contendo todas as características da tabela selecionada (tipo, tamanho, relacionamento, cardinalidade, etc.).

Pacote Service: Os serviços implementam as regras de negócios da PIXEL de acordo com os requisitos da seção 5.2.

Pacote Controller: O *controller* gerencia as funcionalidades definidas nos requisitos funcionais da PIXEL.

Pacote View: A *view* promove a visualização/renderização dos campos e das informações que serão exibidas pela plataforma PIXEL permitindo a interação do usuário.

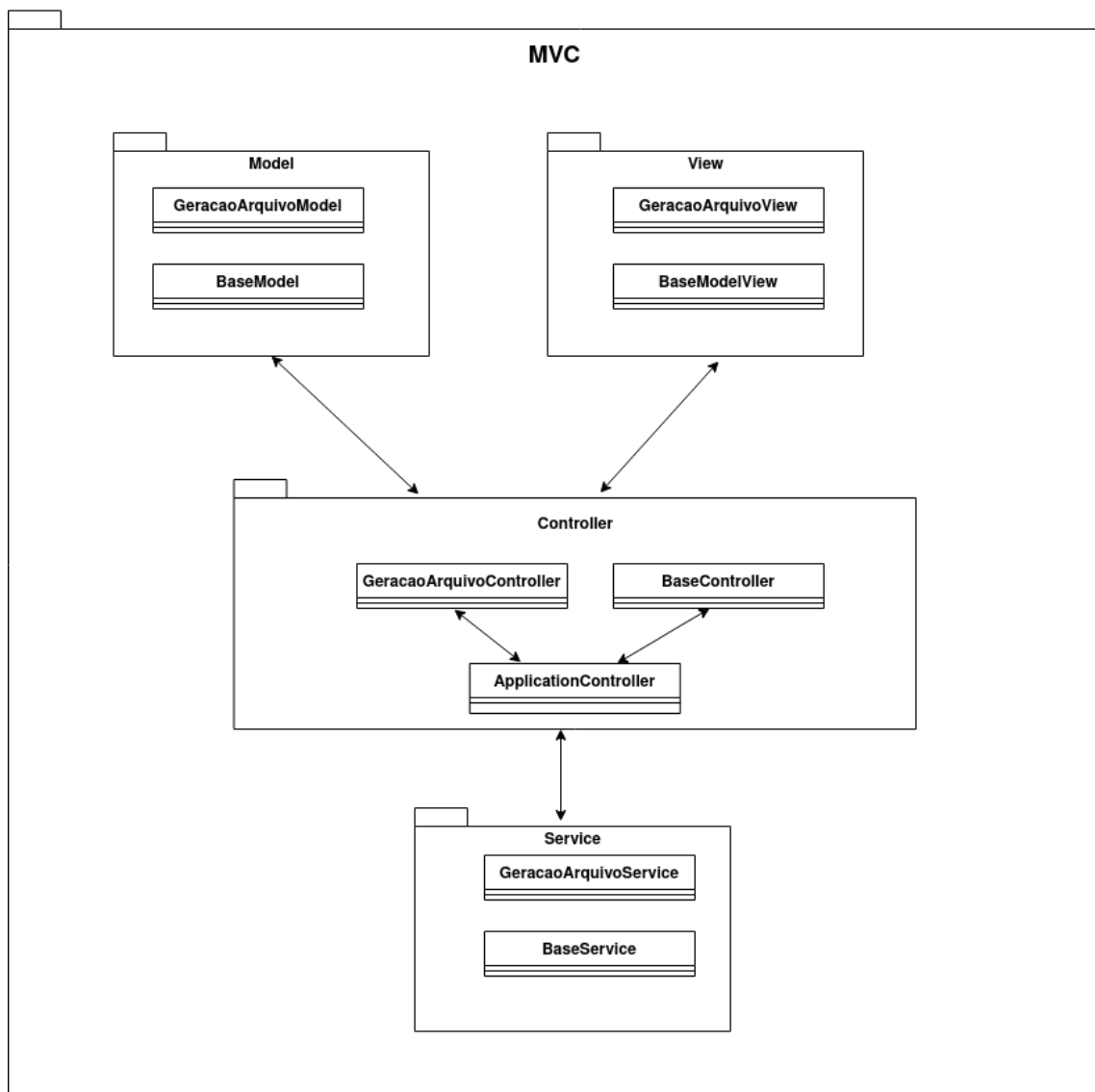
O pacote util está subdividido em três pacotes:

Pacote i18n: O i18n viabiliza a tradução de idiomas das mensagens da PIXEL. Esses idiomas são: Inglês, alemão, espanhol, francês e etc.

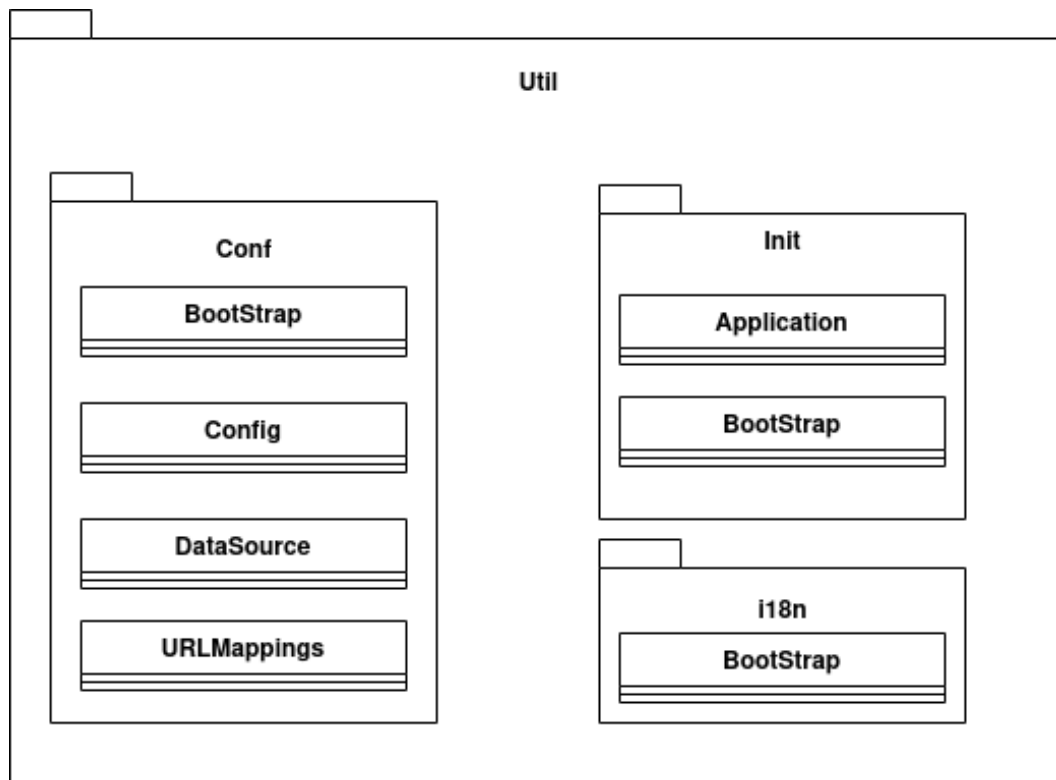
Pacote init: localiza-se as classes para iniciar o grails usando a tecnologia graphQL, visando melhorar o desempenho da PIXEL e viabilizar a abstração de *controllers* que são criadas em tempo de execução na PIXEL.

Pacote conf: Neste pacote estão localizadas as classes responsáveis por realizar as configurações e recursos visuais a serem exibidos na PIXEL, por exemplo: conexão com banco de dados, propriedades da aplicação, gerenciamento dos *endpoints*, etc.

Figura 29 – Diagrama de classes do pacote *MVC*.



Fonte: Próprio Autor.

Figura 30 – Diagrama de classes do pacote *UTIL*.

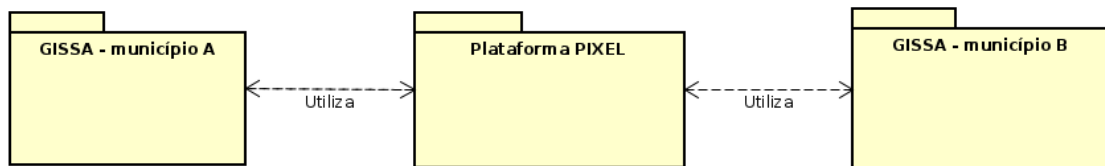
Fonte: Próprio Autor.

Finalizadas as seções sobre a plataforma proposta, no capítulo seguinte tem-se a aplicação da proposta por meio de uma Prova de Conceito usada para validar a PIXEL.

6 Aplicando a plataforma PIXEL em Ambiente Real

Este capítulo apresenta uma PoC para melhor ilustrar as funcionalidades e utilização da PIXEL na tratativa dos aspectos de interoperabilidade em SIS. O cenário utilizado nesta PoC envolve duas instâncias do GISSA devido a disponibilidades em ambiente real e pertinência deste sistema com os objetivos da PoC. O GISSA A (solicitante da informação) e GISSA B (Fornecedor da informação). A PoC foi da plataforma PIXEL foi aplicada em dois municípios gerenciados pelo GISSA, como pode ser visto na Figura 31.

Figura 31 – Diagrama de pacotes dos sistemas participantes.

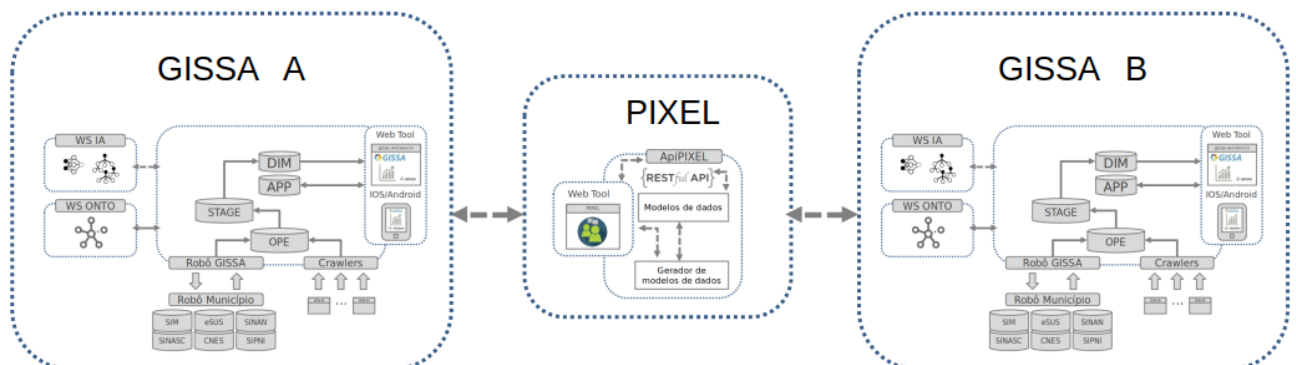


Fonte: Próprio Autor.

- **GISSA A:** Sistema utilizado para receber as informações via PIXEL;
- **PIXEL:** Plataforma utilizada para enviar/receber informações;
- **GISSA B:** Sistema utilizado para enviar as informações via PIXEL.

A Figura 32 é uma explanação conceitual da Figura 31, mostrando o cenário real no qual a simulação realizada. Nela vê-se duas instâncias do GISSA conectadas via PIXEL e sua interface que permitirá a configuração para agregação de serviços do GISSA B no GISSA A.

Figura 32 – Arquitetura GISSA utilizando a PIXEL.



Fonte: Próprio Autor.

A PoC tem início com a identificação do(s) participante(s) no contexto de integração, e com a definição do serviço que se deseja agregar (consumir/disponibilizar informações), fazendo-se uso da interface de configuração da PIXEL, cuja pagina inicial é mostrada na Figura 33. Nela, o usuário deve escolher a base de dados de onde se deseja capturar as informações. Neste caso a base de dados é a robô detalhada na Figura 34.

Ao se selecionar a base de dados (robô), aparecem suas tabelas a serem escolhidas para mapear suas informações via RESTful na base de dados do solicitante (Figura 35). Tendo escolhido a base de dados e a tabela, cria-se o modelo com as informações pré-definidas, como tipo de dados e tamanho. A Figura 36 mostra o modelo criado pela PIXEL na base de dados do solicitante (GISSA A). A Figura 37 exibe o resultado final, ilustrando a troca de informações entre os dois municípios GISSA A e GISSA B).

Figura 33 – Selecionando a base de dados.

Fonte: Próprio Autor.

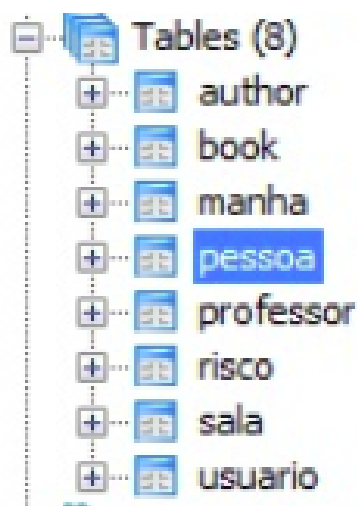
Figura 34 – Selecionando a tabela.

Fonte: Próprio Autor.

Figura 35 – Base de dados e tabela selecionadas.

Fonte: Próprio Autor.

Figura 36 – Criação do modelo.



Fonte: Próprio Autor.

Figura 37 – Integração dos Sistemas.

Nome:				
Todos				
Filtrar				
Exportar				
Dados				
Id	Nome	Sobrenome	Idade	
1	Gomes	Leonardo	25	
2	Castro	Paulo	40	
3	Cambriza	Jose	60	
4	Danilov	Cátia	70	
5	Soldanka	Marcon	20	
6	Sosa	Leonardo	23	
7	Araujo	Milena	40	
8	Vieira	Estebanobe	60	
9	Alves	Rodrigo	70	
10	Gomes	Juanes	20	
Total de registros: 10				
< 1 >				

Fonte: Próprio Autor.

A Figura 37 mostra o sistema GISSA A, que, anteriormente a aplicação da PIXEL, não possuía estas informações da tabela pessoa (id, nome, sobrenome, idade) escolhida dentre as disponíveis no GISSA B.

Com a PIXEL, foi possível utilizar estas informações possibilitando a interoperabilidade dos níveis fundamental e sintática, via *API RESTful* entre os sistemas envolvidos.

7 Conclusão

O estudo de interoperabilidade e a experiência prática no desenvolvimento de SIS possibilitou a definição e implementação da PIXEL, uma plataforma interoperável para sistemas de saúde legados ou com tecnologia proprietária, viabilizando a integração de novos serviços, por vezes difíceis, senão impossíveis, devido à falta de documentação e/ou de profissionais especializados em tecnologias ultrapassadas/ proprietárias.

Embora a PIXEL tenha sido elaborada e testada no contexto do GISSA, um ambiente de tecnologia proprietária, sua concepção é plenamente adaptável a qualquer outro SIS. A inovação de proposta do PIXEL em seu aspecto fundamental e sintático está na estratégia de criação de modelos, a partir da ideia de um gerador de modelos e da abordagem dinâmica de criação de *endpoints* e serviços via *API RESTful*.

Esta proposta foi objeto de uma PoC que teve sua validação na geração de um novo relatório em um ambiente experimental (GISSA A) a partir de dados consumidos de um outro sistema (GISSA B). As *Uniform Resource Locators* (URLs) da PIXEL são todas dinâmicas e flexíveis a mudanças, possibilitando sua evolução para uma estrutura mais complexa e com maior potencial de soluções de interoperabilidades.

Assim, nesta PoC não foi implementada a interoperabilidade semântica permitida pela PIXEL, via o componente GIRLS. Embora este componente esteja implementado, os ambientes A e B do GISSA não tratam (ainda) este problema. Em um cenário futuro para esta PoC semântica da PIXEL, poder-se-ia imaginar empresas cujo SIS usa o FHIR como padrão (Unimed, por exemplo) no lugar da plataforma GISSA, tentando interoperar com algum sistema governamental que tenha adotado o OpenEHR.

Vale destacar que a plataforma PIXEL incrementa uma importante funcionalidade ao GISSA. Sem a PIXEL, qualquer integração de novos serviços exige do implementador um profundo conhecimento das funcionalidades e aspectos de implementação da plataforma AWF, utilizada pelo GISSA para operar seus robôs na captura de informação nas bases de dados e posterior tratamento para geração de relatórios, alertas, gráficos, etc. Como discutido na PoC, apresentada no Capítulo 6, o uso da PIXEL permitiu, sem conhecimentos específicos da plataforma AWF, que o um sistema (GISSA A) agregasse tabelas (*endpoints*) de um sistema externo aos seu ambiente (GISSA B).

7.1 Limitações do Trabalho

Durante a produção deste trabalho algumas limitações foram identificadas. Uma delas refere-se à implementação do gerador de modelos usando apenas tabelas para

representá-los. Neste aspecto, a representação dos modelos limita-se ao uso de tabelas, por meio dos arquivos .sqls. Entretanto, como trabalhos futuros para esta questão, pretende-se consolidar um gerador de modelos com maior nível de abstração que inclua suporte a modelos usando arquivos xml, *JavaScript Object Notation* (JSON) e Texto (txt).

Outra limitação do trabalho, como já comentado, foi a utilização da plataforma apenas entre simulações do GISSA, não permitindo o sistema ser validado em bancos de dados de paradigmas diferentes. Entretanto, o objetivo de criar modelos permitiu mostrar a utilização da plataforma proposta e sua potencialidade para facilitar a interoperabilidade.

7.2 Trabalhos Futuros

Como trabalhos futuros sugere-se a validação integral da PIXEL, seja em um gerador de modelos que possa contemplar mais recursos, quanto no aspecto semântico que não foi testado. Pretende-se realizar a implementação do mecanismo *from-to* (De-para) para tornar os campos mais dinâmicos e suscetíveis a mudanças constantes, tunelamento de carga visando obter a passagem de alta quantidade de informações intermediada pela plataforma e a criação visual do cenário de integração supervisionado por BPMN para acompanhar o processo de integração dos sistemas. Desse modo, será possível enriquecer A PIXEL, adaptando-a a possíveis problemas de integração que vão surgindo, em diferentes níveis de interoperabilidade.

Há, naturalmente, uma expectativa de que este trabalho seja incorporado ao produto GISSA, ambiente sem o qual as ideias e motivações que conceberam e implementaram o PIXEL não teriam razão de ser.

Referências

- ABUGABAH, A. Evaluation of healthcare enterprise information systems: A structural equation model. In: IEEE. *2017 International Conference on Information and Digital Technologies (IDT)*. [S.l.], 2017. p. 1–4. Citado na página 34.
- AMARAL, L. *PRAXIS: Um referencial para o Planeamento de Sistemas de Informação*. Tese (Doutorado) — Universidade do Minho, 1994. Citado na página 33.
- AMIN, M. M. et al. Interoperability framework for integrated e-health services. *Bulletin of Electrical Engineering and Informatics*, v. 9, n. 1, p. 354–361, 2020. Citado na página 42.
- ANDRADE, L.; OLIVEIRA, M.; RAMOS, R. *Projeto GISSA: META FÍSICA 3–atividade 3.1 Definir modelo de inteligência de gestão na saúde*. 2015. Citado 2 vezes nas páginas 28 e 36.
- ARMS, W. A. et al. A spectrum of interoperability, the site for science prototype for the nsdl. *D-Lib magazine; 2002 (8) 1*, Corporation For National Research Initiatives, 2002. Citado na página 33.
- BAKAR, H. K. A.; RAZALI, R.; JAMBARI, D. I. Implementation phases in modernisation of legacy systems. In: IEEE. *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*. [S.l.], 2019. p. 1–6. Citado na página 35.
- BEIR, M. A. d. *Arquiteturas para interoperabilidade de sistemas de informação na área da saúde: caso de demonstração-prescrição eletrónica de medicamentos*. Tese (Doutorado) — Universidade do Minho, 2018. Citado na página 43.
- BISHR, Y. Semantic aspects of interoperable gis. In: ITC. [S.l.], 1997. Citado na página 31.
- BRODIE, M. L.; STONEBRAKER, M. Legacy information systems migration: Gateways, interfaces, and the incremental approach. Morgan Kaufmann Publishers Inc., 1995. Citado na página 35.
- CHU, S. C. From component-based to service oriented software architecture for healthcare. In: IEEE. *Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005*. [S.l.], 2005. p. 96–100. Citado na página 33.
- CRUZ-CORREIA, R. et al. ehealth key issues in portuguese public hospitals. In: IEEE. *2012 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS)*. [S.l.], 2012. p. 1–6. Citado na página 33.
- CRUZ, M. M. da et al. Semanticsus: Um portal semântico baseado em ontologias e dados interligados para acesso, integração e visualização de dados do sus. In: *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*. Porto Alegre, RS, Brasil: SBC, 2019. p. 13–18. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbcas_estendido/article/view/6277>. Citado 2 vezes nas páginas 47 e 48.

- FARINELLI, F.; ALMEIDA, M. B. Interoperabilidade semântica em sistemas de informação de saúde por meio de ontologias formais e informais: um estudo da norma openehr. In: *Conferência Biredial Istec*. [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 25 e 43.
- FILHO, W. de P. P. *Engenharia de software*. [S.l.]: LTC, 2003. v. 2. Citado na página 35.
- FOWLER, M. Richardson maturity model: steps toward the glory of rest. *Online at <http://martinfowler.com/articles/richardsonMaturityModel.html>*, p. 24–65, 2010. Citado na página 51.
- GESSERT, F. et al. Towards a scalable and unified rest api for cloud data stores. In: *GI-Jahrestagung*. [S.l.: s.n.], 2014. p. 723–734. Citado na página 41.
- GOMES, F. et al. Girls, a gateway for interoperability of electronic health record in low-cost system:* interoperability between flhir and openehr standards. In: *IEEE. 2019 IEEE International Conference on E-health Networking, Application & Services (HealthCom)*. [S.l.], 2019. p. 1–6. Citado 3 vezes nas páginas 50, 52 e 56.
- HENNING, F. Adoption of interoperability standards in government information networks: an initial framework of influence factors. In: *ACM. Proceedings of the 7th International Conference on Theory and Practice of Electronic Governance*. [S.l.], 2013. p. 264–267. Citado na página 25.
- JAY, F.; GOETZ, J. Ieee standard dictionary of electrical and electronics terms. In: *INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS*. [S.l.], 1988. Citado na página 26.
- JAYARATNE, M. et al. A data integration platform for patient-centered e-healthcare and clinical decision support. *Future Generation Computer Systems*, Elsevier, v. 92, p. 996–1008, 2019. Citado na página 42.
- JOHANN, S. Dave thomas on innovating legacy systems. *IEEE Annals of the History of Computing*, IEEE Computer Society, v. 33, n. 02, p. 105–108, 2016. Citado na página 35.
- LIMA, J. d. S. et al. Uma arquitetura de software para implementação de um ehr utilizando soa considerando a interoperabilidade entre sistemas legados. Universidade Federal de Sergipe, 2016. Citado 2 vezes nas páginas 35 e 43.
- LOPES, G.; VIDAL, V.; OLIVEIRA, M. A framework for creation of linked data mashups: A case study on healthcare. In: *ACM. Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web*. [S.l.], 2016. p. 327–330. Citado 2 vezes nas páginas 28 e 36.
- MAHMOUD, K. M. A unified messaging-based architectural pattern for building scalable enterprise service bus. In: *ACM. Proceedings of International Conference on Information Integration and Web-based Applications & Services*. [S.l.], 2013. p. 697. Citado na página 25.
- MARINOS, A.; WILDE, E.; LU, J. Http database connector (hdbc) restful access to relational databases. In: *Proceedings of the 19th international conference on World wide web*. [S.l.: s.n.], 2010. p. 1157–1158. Citado na página 41.
- MILLER, P. Interoperability: What is it and why should i want it? *Ariadne*, n. 24, 2000. Citado na página 31.

- MIYOSHI, N. S. B. et al. An ehealth platform for the support of a brazilian regional network of mental health care (ehealth-interop): Development of an interoperability platform for mental care integration. *JMIR Mental Health*, JMIR Publications Inc., Toronto, Canada, v. 5, n. 4, p. e10129, 2018. Citado na página 42.
- MORAIS, R. M.; GOMES, E.; COSTA, A. Os sistemas de informação do sus: uma perspectiva histórica e as políticas de informação e informática. *Nucleus*, v. 11, n. 1, p. 287–304, 2014. Citado na página 26.
- NASCIMENTO, S. C. do; CARVALHO, F. O.; ROCHA, T. da. Um framework para interoperabilidade entre componentes distribuídos heterogêneos. *Revista Brasileira de Administração Científica*, v. 4, n. 2, p. 239–256, 2013. Citado na página 42.
- OLIVEIRA, M. et al. A context-aware framework for health care governance decision-making systems: A model based on the brazilian digital tv. In: IEEE. *2010 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. [S.l.], 2010. p. 1–6. Citado 2 vezes nas páginas 28 e 36.
- PERUMAL, T.; RAMLI, A. R.; LEONG, C. Y. Interoperability framework for smart home systems. *IEEE Transactions on Consumer Electronics*, IEEE, v. 57, n. 4, p. 1607–1611, 2011. Citado na página 42.
- POLÁK, M.; HOLUBOVÁ, I. Rest api management and evolution using mda. In: *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*. [S.l.: s.n.], 2015. p. 102–109. Citado na página 41.
- PRESS, C. Boca raton, 2004. *There is no corresponding record for this reference.*[Google Scholar], 2010. Citado na página 27.
- RIBEIRO, L. F. d. S. Interoperabilidade nos sistemas de informação de saúde-das convicções à realidade. Faculdade de Medicina da Universidade do Porto, 2011. Citado na página 25.
- RODRÍGUEZ-ECHEVERRÍA, R. et al. Model-driven generation of a rest api from a legacy web application. In: SPRINGER. *International Conference on Web Engineering*. [S.l.], 2013. p. 133–147. Citado na página 41.
- SANTOS, E. M. dos. Implementing interoperability standards for electronic government: An exploratory case study of the e-ping brazilian framework. *International Journal of Electronic Government Research (IJEGR)*, IGI Global, v. 4, n. 3, p. 103–112, 2008. Citado na página 31.
- SANTOS, E. M. dos. A adoção da arquitetura e-ping: Um estudo de caso na fiocruz/bahia. 2011. Citado na página 31.
- SARIPALLE, R. K. Fast health interoperability resources (fhir): Current status in the healthcare system. *International Journal of E-Health and Medical Communications (IJEHMC)*, IGI Global, v. 10, n. 1, p. 76–93, (2019). Citado na página 43.
- SAÚDE, S. Atenção à. Conjunto mínimo de dados da atenção à saúde. In: SUS. *I Encontro Nacional do Conjunto Mínimo de Dados da Atenção à Saúde*. [S.l.], 2015. Citado na página 25.

- SAYÃO, L. F.; MARCONDES, C. H. O desafio da interoperabilidade e as novas perspectivas para as bibliotecas digitais. *Transinformação*, Pontifícia Universidade Católica de Campinas, v. 20, n. 2, p. 133–148, 2008. Citado 2 vezes nas páginas 31 e 33.
- SELLAMI, R.; BHIRI, S.; DEFUDE, B. Odbapi: a unified rest api for relational and nosql data stores. In: IEEE. *2014 IEEE International Congress on Big Data*. [S.l.], 2014. p. 653–660. Citado na página 41.
- SHETH, A. P. Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In: *Interoperating geographic information systems*. [S.l.]: Springer, 1999. p. 5–29. Citado 2 vezes nas páginas 31 e 32.
- SILSAND, L. The ‘holy grail’ of interoperability of health information systems: Challenges and implications. In: SPRINGER. *Scandinavian Conference on Information Systems*. [S.l.], 2017. p. 140–154. Citado na página 43.
- SILVA, C. et al. Laís, um analisador baseado em classificadores para a geração de alertas inteligentes em saúde. In: SBC. *Anais do I Workshop de Computação Urbana (COURB 2017)*. [S.l.], 2017. v. 1, n. 1/2017. Citado 2 vezes nas páginas 28 e 36.
- SOARES, H. A.; MOURA, R. S. A methodology to guide writing software requirements specification document. In: IEEE. *2015 Latin American Computing Conference (CLEI)*. [S.l.], 2015. p. 1–11. Citado na página 56.
- UMAR, A. *Application (re) engineering: building Web-based applications and dealing with legacies*. [S.l.]: Prentice Hall Press, 1997. Citado na página 35.
- VERNADAT, F. B. Interoperable enterprise systems: Principles, concepts, and methods. *Annual reviews in Control*, Elsevier, v. 31, n. 1, p. 137–145, 2007. Citado na página 31.
- WHITE, K. L. Information for health care: an epidemiological perspective. *Inquiry*, JSTOR, v. 17, n. 4, p. 296–312, 1980. Citado na página 34.
- ZEINALI, N.; ASOSHEH, A.; SETAREH, S. The conceptual model to solve the problem of interoperability in health information systems. In: IEEE. *2016 8th International Symposium on Telecommunications (IST)*. [S.l.], 2016. p. 684–689. Citado na página 34.
- ZHANG, G.; DU, Y.; LI, G. Evaluation system of education quality in the evolution of information technology. In: IEEE. *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*. [S.l.], 2011. p. 4064–4067. Citado na página 33.