Ministério da Justiça

Projeto: SISDEPEN

Nota Técnica

Revisão	Descrição	Autor	Data
1.0	Construção do documento	Israel Branco	28/08/2020

1 Sumário

2 Introdução

SISDEPEN é o sistema de gestão de custodiados do sistema penitenciário federal e este documento visa reportar o resultado da análise efetuada na aplicação. Para este estudo foram desconsiderados todo o contexto negocial ao qual a ferramenta está inserida, também foram desconsideradas o ambiente ao qual a ferramenta esta operando sendo analisado puramente questões que tangem a qualidade de código, padrões de codificação, vulnerabilidades de dependências, modelo relacional de banco de dados e concepção arquitetural.

Este sistema foi construído pela empresa SERPRO e possui boa documentação de suas funcionalidades a nível de usuário, negocial e arquitetura. Para fins de armazenamento e disponibilidade, estes documentos estão disponíveis no repositório git https://gitlab.mj.gov.br/cgsis/sisdepen/tree/master/documentos. O código fonte da aplicação está disponível em https://gitlab.mj.gov.br/cgsis/sisdepen/tree/master/codigo_fonte e as configurações necessárias para os servidores de aplicação em https://gitlab.mj.gov.br/cgsis/sisdepen/tree/master/jboss-dsv.

3 Apresentação do cenário atual

Esta sessão ira descrever a arquitetura, tecnologias, frameworks e dependências que compõe a base da aplicação.

O SISDEPEN está construído para funcionar em ambiente WEB com uma pequena parte voltada para mobile, possui segregação entre as camadas de front-end/back-end e sua arquitetura esta projetada para trabalhar de forma desacoplada e distribuída.

O backend da aplicação está construído sobre a stack Java Enterprise Edition 6, já a aplicação front-end está construída para trabalhar com SPA – Single Page Aplication utilizando o framework Angular JS.

Os diagramas a seguir representam o modelo de componentes ao qual a aplicação está construída, suas dependências, fluxo sequencial e seu modelo de comunicação entre os módulos.

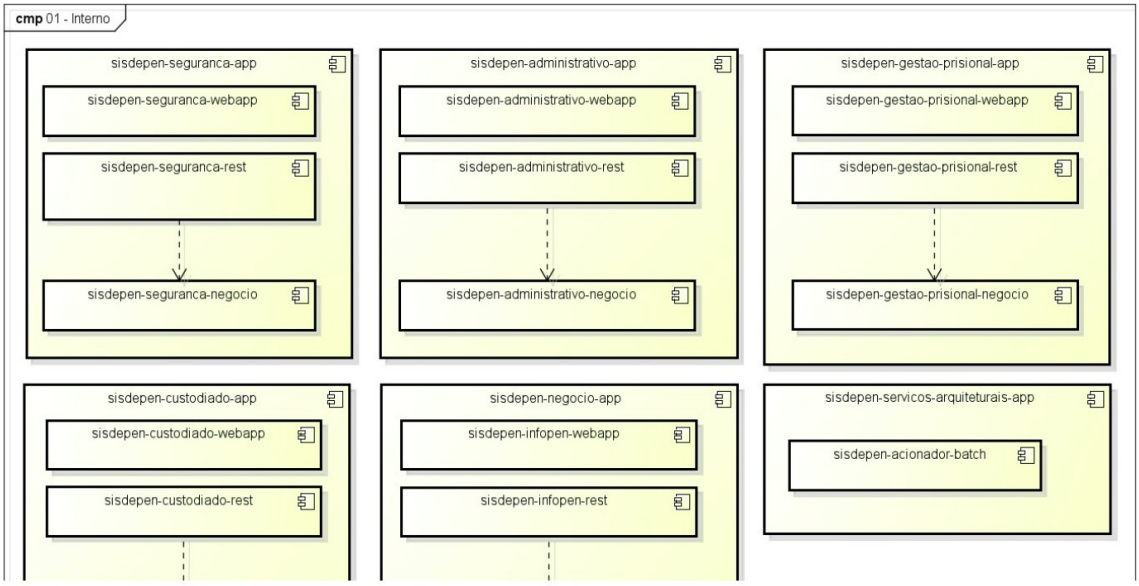

Figura 1: empacotamento dos componentes

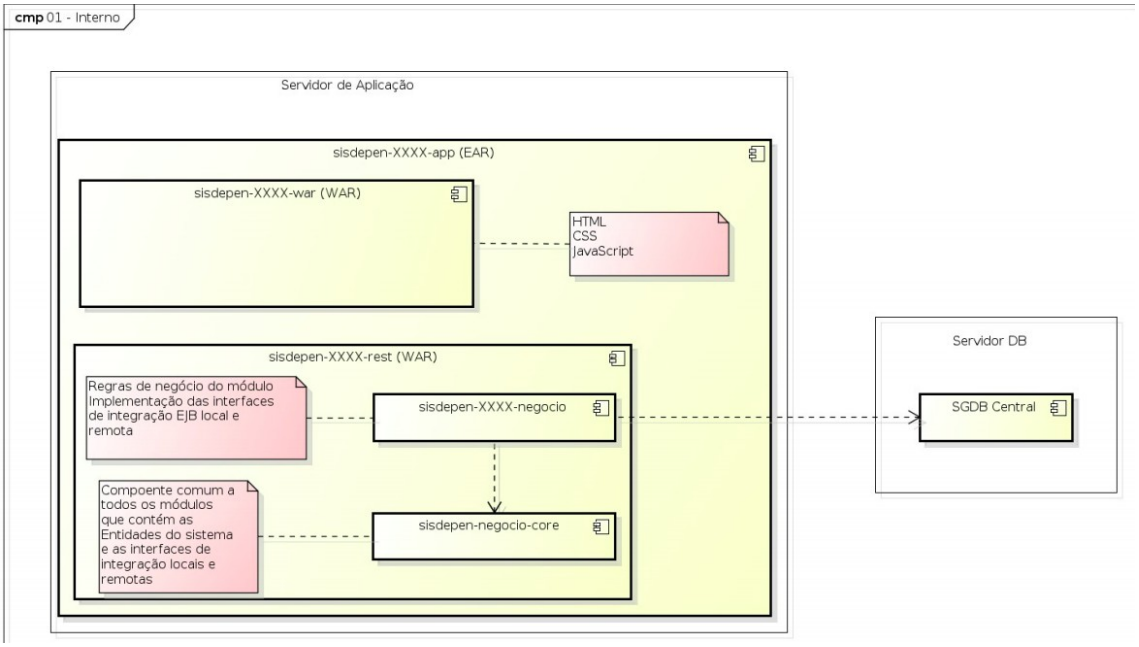

Figura 2: estrutura de componentes

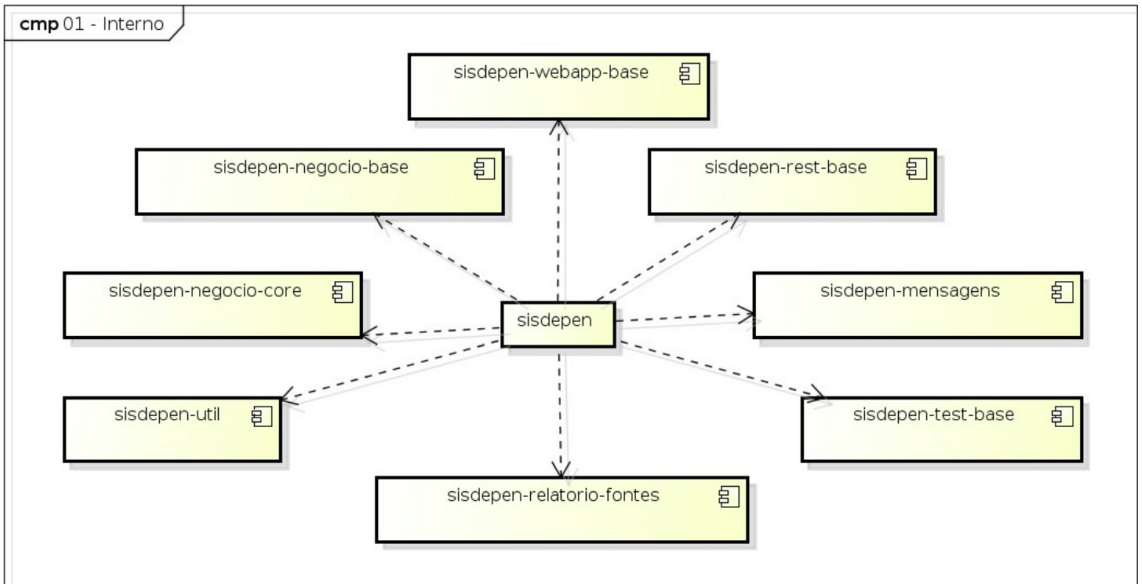

Figura 3: componentes base

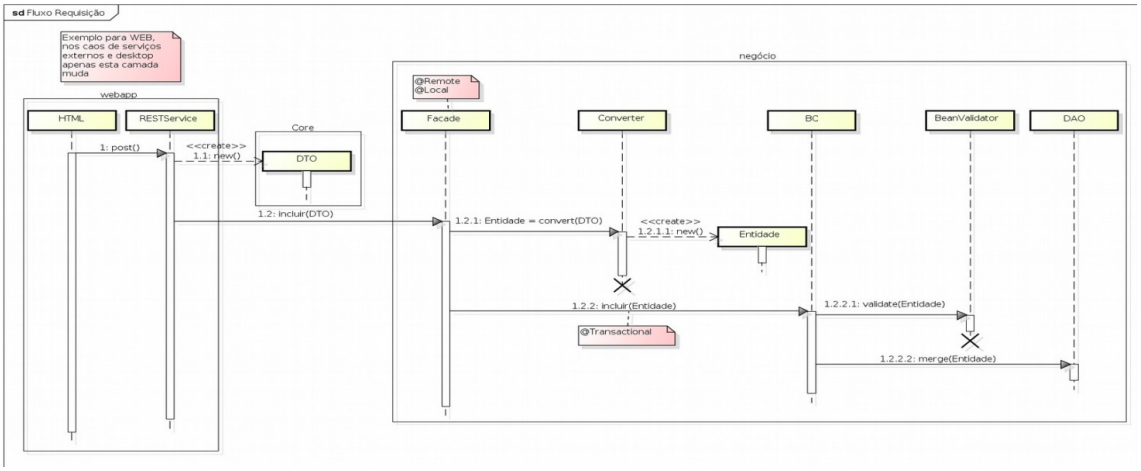

Figura 4: fluxo principal

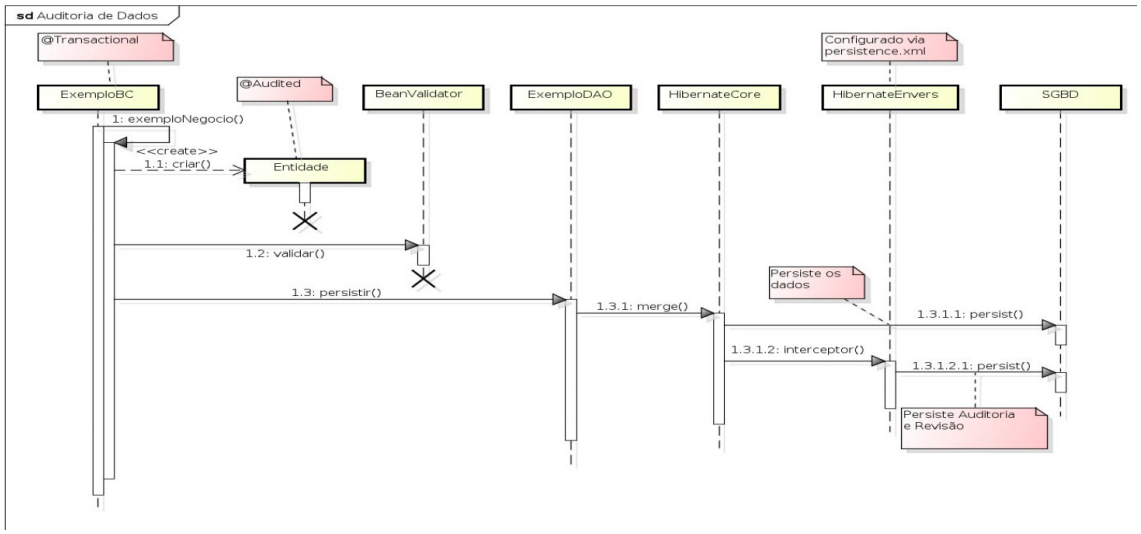

Figura 5: fluxo de auditoria de dados

3.1 Módulos

A aplicação está composta por 5 módulos, sendo eles: Sisdepen Administrativo, Sisdepen Custodiado, Sisdepen Gestão prisional, Sisdepen Infopen, Sisdepen Segurança. Todos os módulos funcionam de forma independente e desacoplada, toda comunicação entre os mesmos são feitas por intermédio de interfaces remotas utilizando a especificação EJB 3.

A organização dos módulos segue a mesma estrutura de empacotamento e lógica de organização. A figura a seguir é refente ao módulo administrativo, contudo a mesma analogia é aplicada aos demais módulos da aplicação.

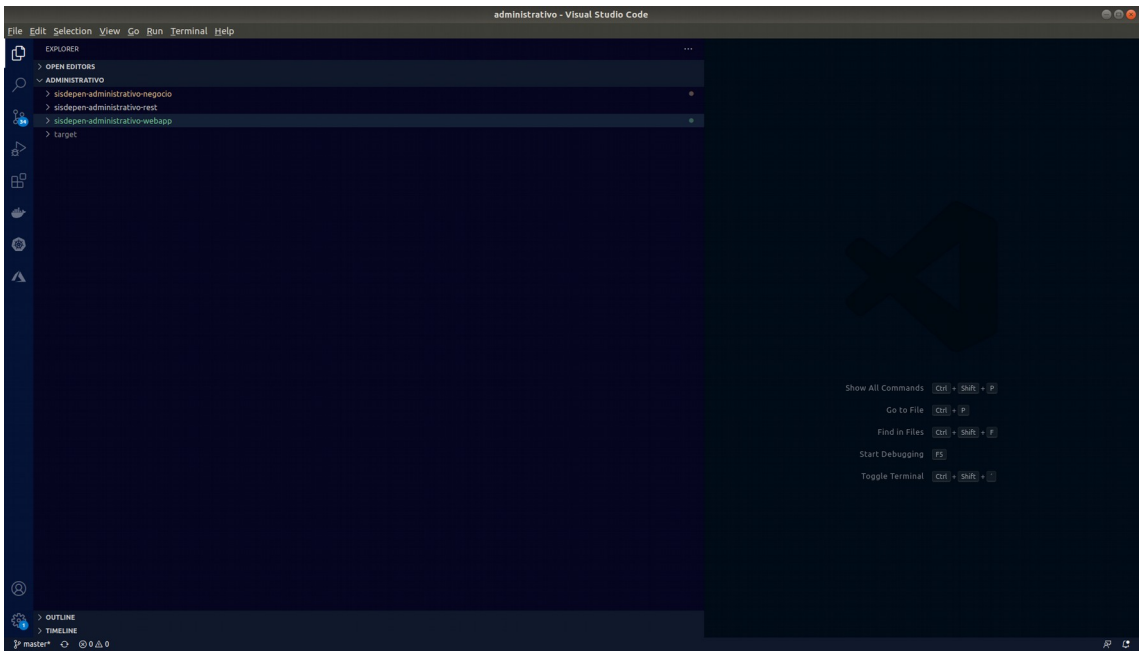

Figura 6: Estrutura dos módulos

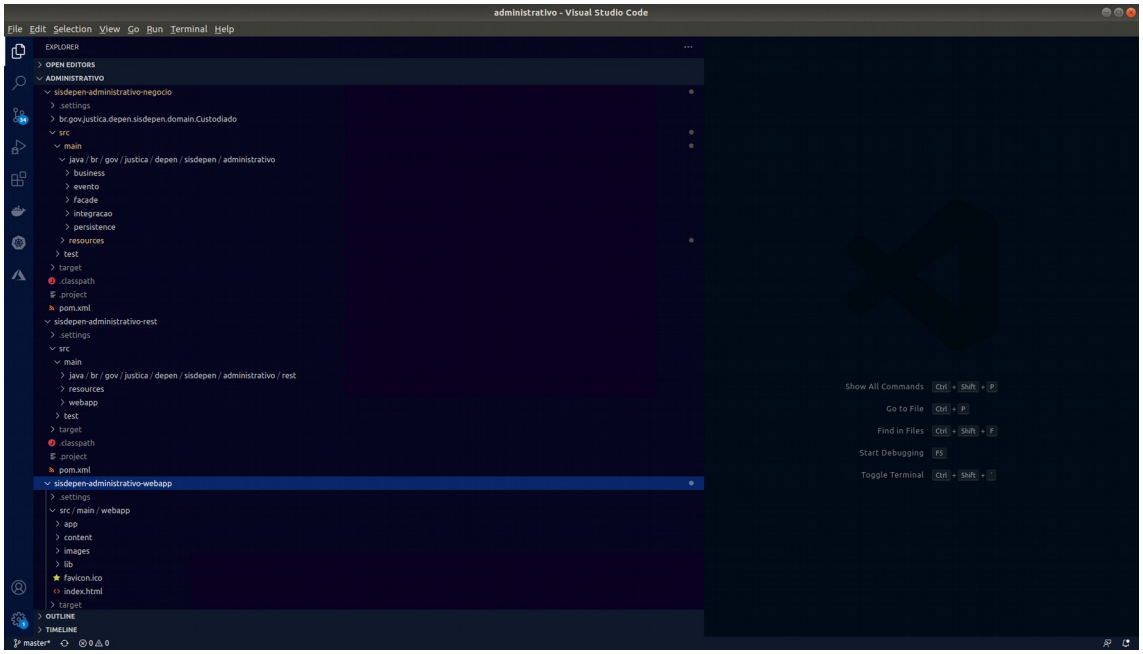

Figura 7: Estrutura de empacotamento

3.2 Tecnologias utilizadas

Esta sessão descreve as tecnologias, frameworks e principais bibliotecas utilizadas na construção do projeto, descrevendo versões e propósitos de utilização.

Nome	Versão	Utilização	Observação
Java	1.8	Linguagem de programação.	
Angular	JS	Framework Web.	
Hibernate	x	Framework ORM.	
Hibernate Search	x	Componente de busca por aproximação textual.	
EJB	3	Componente corporativo para execução de forma local, distribuída e transacional.	
Demoiselle	2.5	Framework desenvolvido pelo SERPRO para padronização no desenvolvimento de aplicações.	
Jboss EAP	6.4	Servidor de aplicação	
NodeJs	4.6.0	Javascript runtime	
Oracle	10g	Servidor de banco de dados.	

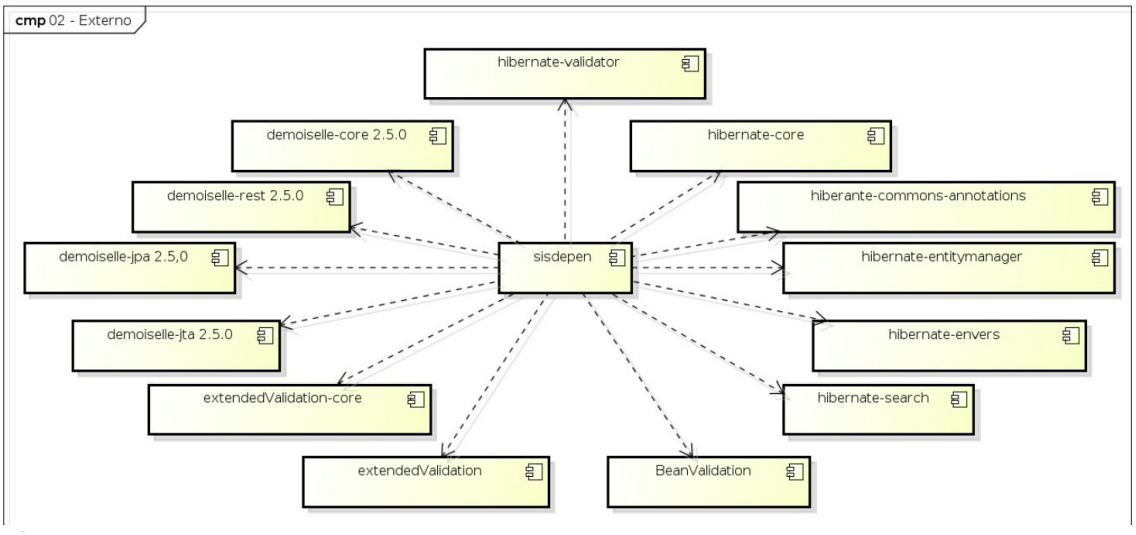

Figura 8: principais dependências

3.3 Modelagem de dados

A estrutura de banco de dados esta composta pela utilização de 3 schemas (sisdepen, sisdepen_app, sisdepen_aud) e são estes responsáveis por armazenar as informações transacionais dos módulos que compõe a aplicação, da aplicação mobile e dados de auditoria respectivamente.

Sendo o schema sisdepen o core da aplicação, as imagens a seguir representam os principais relacionamentos. A modelagem completa juntamente com o dicionário de dados estão disponíveis em https://gitlab.mj.gov.br/cgsis/sisdepen/tree/master/documentos/modelo%20de%20dados.

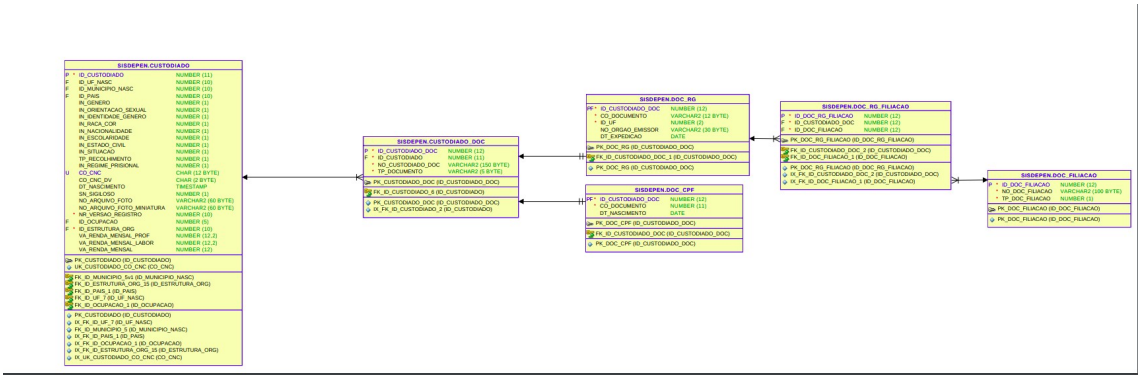

Figura 9: modelagem - custodiado

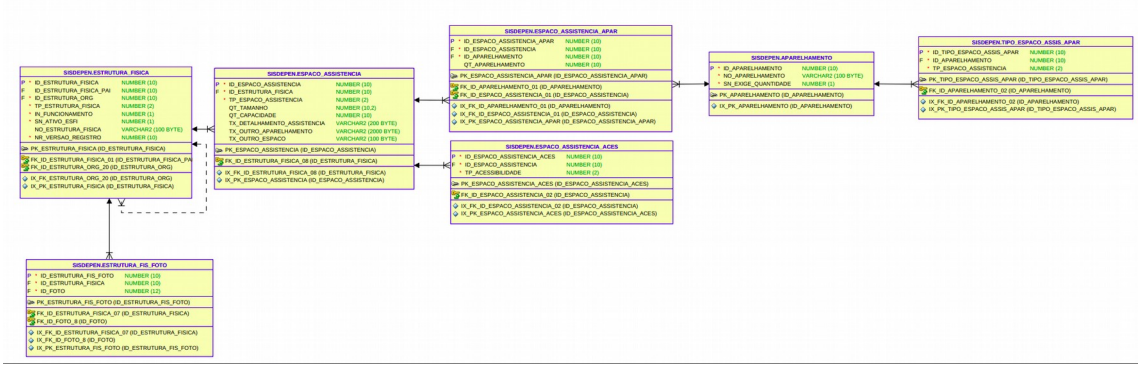

Figura 10: modelagem - assistência

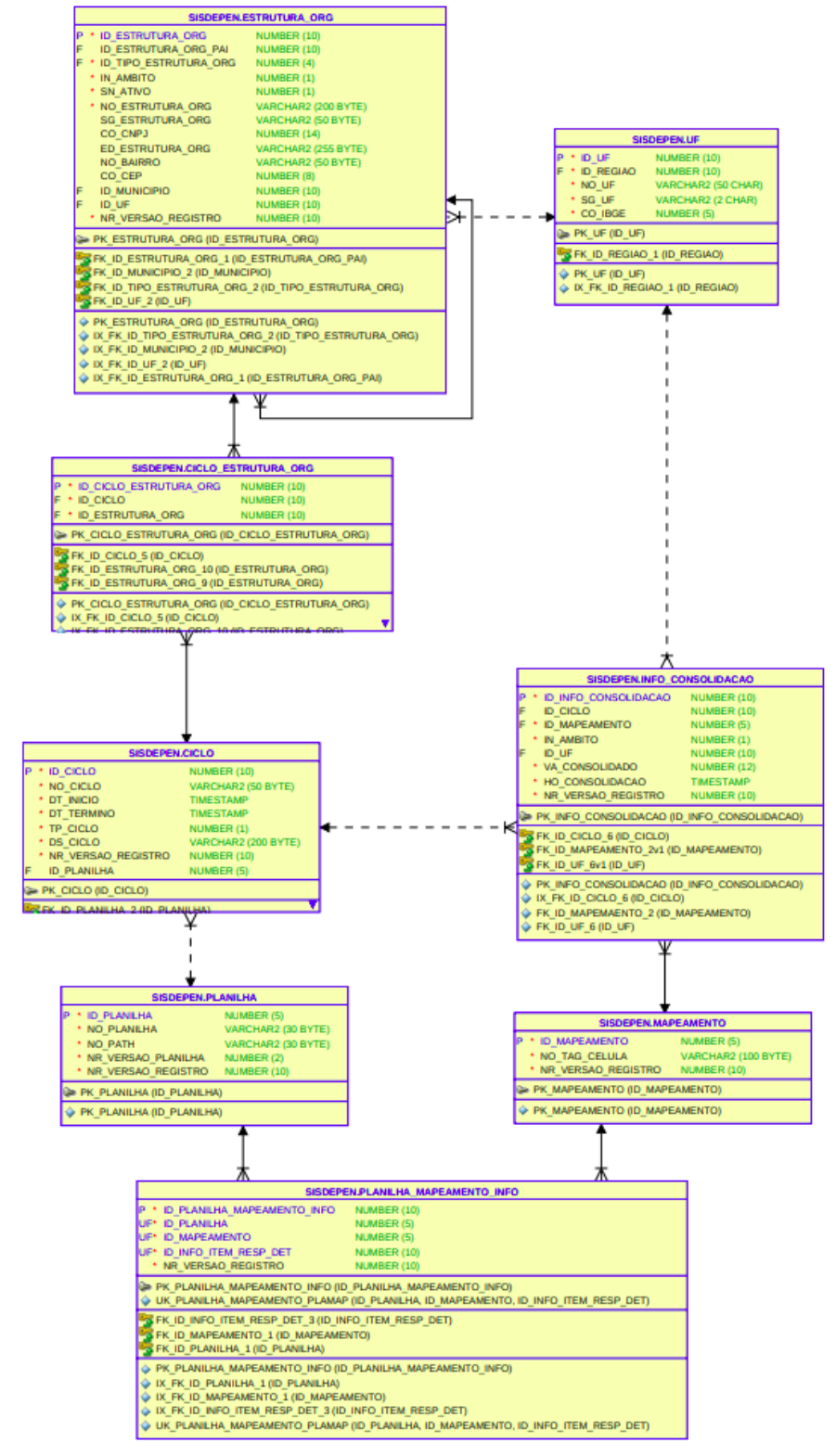

Figura 11: modelagem - consolidação

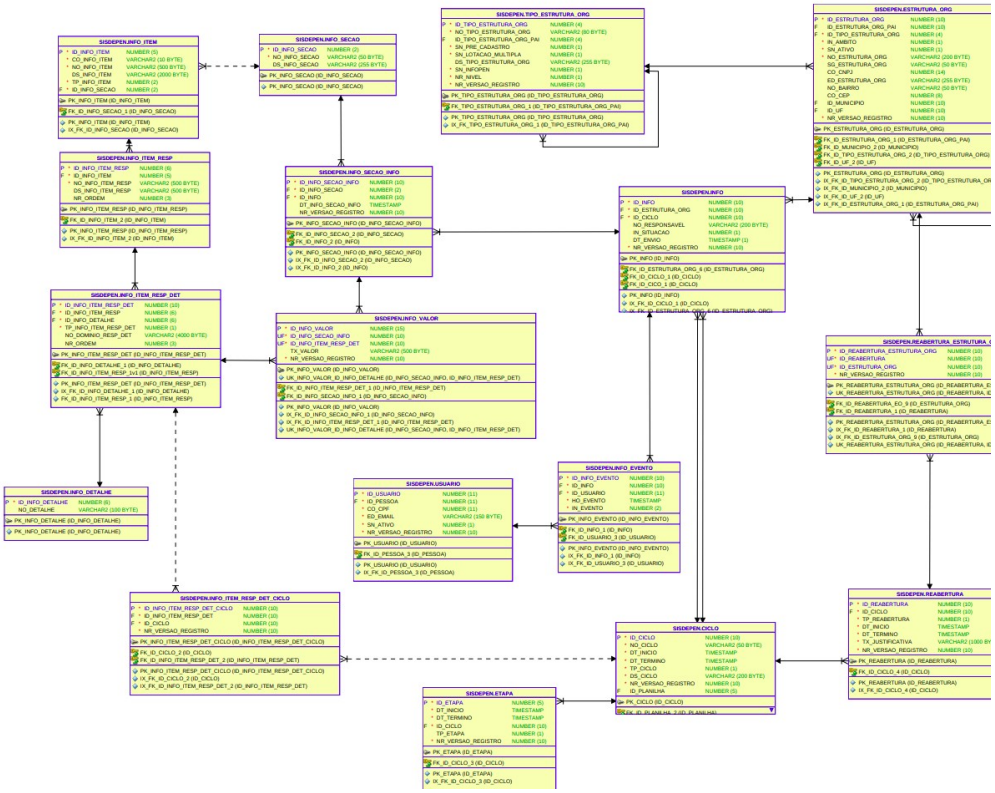

Figura 12: modelagem - infopen

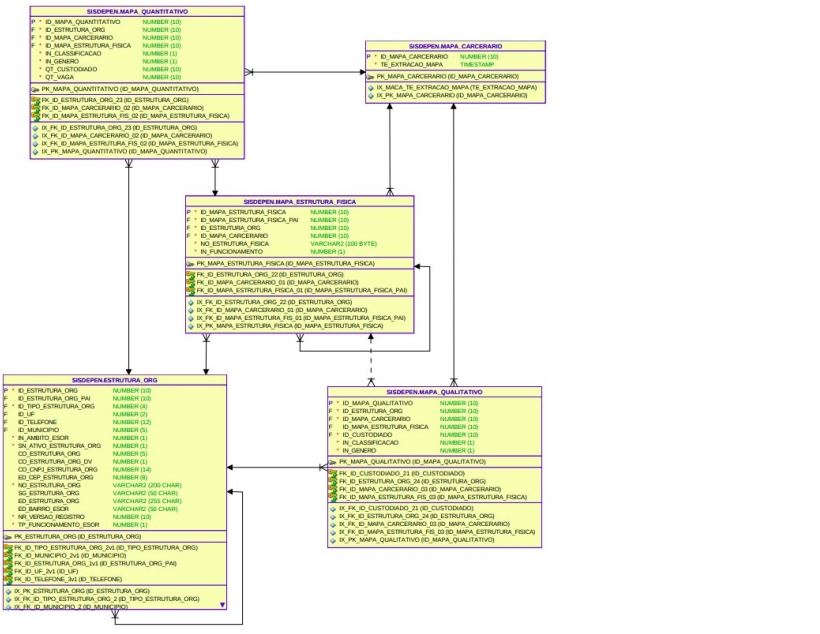

Figura 13: modelagem - mapa carcerário

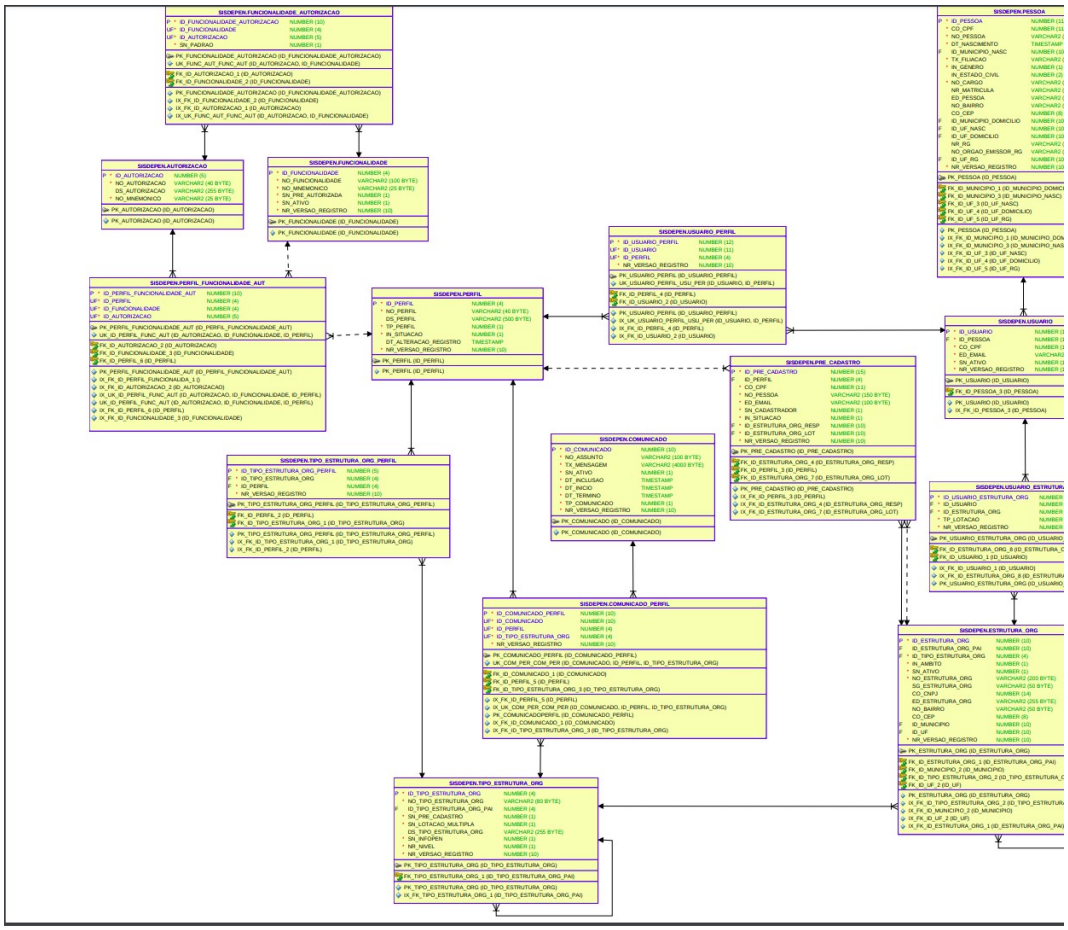

Figura 14: modelagem - usuário

4 Análise técnica

Este tópico descreve a ferramenta do ponto de vista técnico, tanto nos aspectos de codificação, análise estática de código, análise de vulnerabilidade de dependências e particularidades de implementação.

4.1 SonarQube

Ferramenta utilizada para verificação de estática de código. Para esta análise não foram utilizadas as métricas de qualidade implantadas no SonarQube do Ministério da Justiça, contudo foram utilizadas as regras padrões de análise da ferramenta. Os resultados foram os seguintes para os componentes utilitários, componentes negociais e componentes da camada rest das módulos:

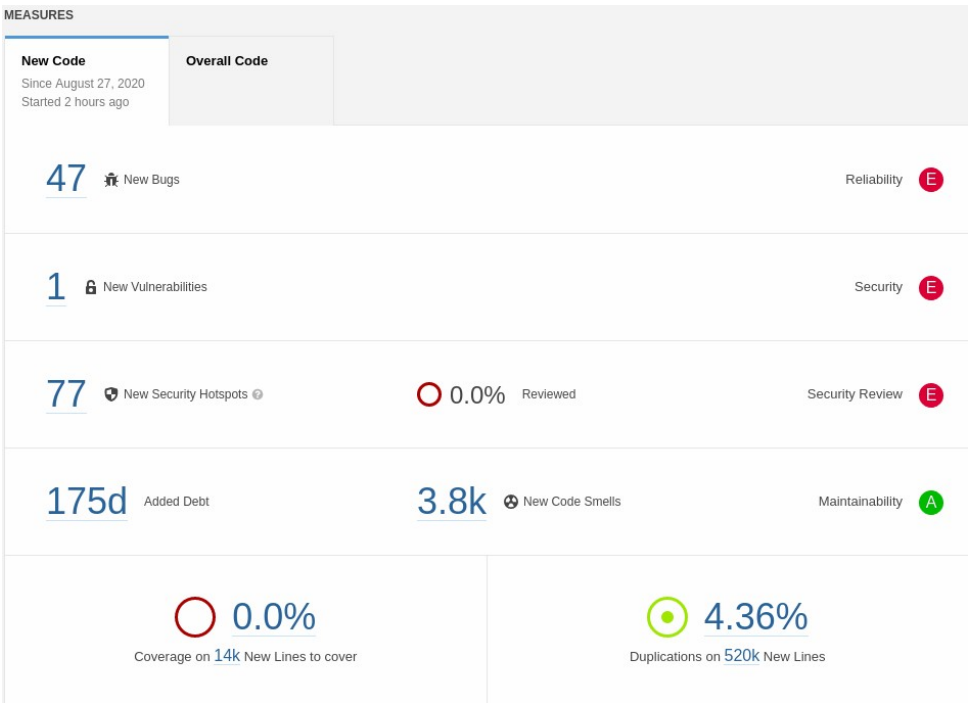

Figura 15: Análise estática de código

Para a obtenção dos resultados, fora utilizado o código fonte referente a tag 6.13.10 https://gitlab.mj.gov.br/cgsis/sisdepen/-/tags/6.13.10:

- 47 bugs;
- 77 violações de segurança;
- 3.8 mil violações de más práticas (complexidade cognitiva, complexidade ciclomática, débito técnico e outros);
- 4.36% de duplicação de código;

O relatório da ferramenta Sonarqube apresenta 0% de cobertura de testes tendo em vista dado a falta de integração com a ferramenta Jacoco para análise de cobertura de testes, contudo percebe-se a presença de boa cobertura de testes (próximo aos 100%) na camada de serviço da aplicação .

4.2 OWASP Dependency Check

A utilização de bibliotecas de terceiros aumenta substancialmente a produtividade na construção de um software, contudo estas podem trazer consigo vulnerabilidades que afetam diretamente a segurança da aplicação. A ferramenta Dependency Check tem como propósito efetuar análise de vulnerabilidade de dependências utilizadas no projeto back-end, a seguir temos as principais informações extraídas desta análise.

servicos-bilhetagem				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
log4j-1.2.17.jar	CRITICAL	2	Highest	29
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
commons-beanutils-1.9.2.jar	HIGH	1	Highest	37
jackson-databind-2.1.4.jar	CRITICAL	39	Highest	38
poi-3.13.jar	HIGH	4	Highest	29
guava-14.0.1.jar	MEDIUM	1	Highest	21
weld-core-1.1.28.Final.jar	MEDIUM	1		27
slf4j-ext-1.7.2.jar	CRITICAL	1	Highest	29
cxf-rt-ws-security-2.7.14.jar	HIGH	9	Highest	39
cxf-rt-core-2.7.14.jar	HIGH	8	Highest	41
opensaml-2.6.1.jar	HIGH	3	Highest	43
serializer-2.7.1.jar	HIGH	1	Low	31
wss4j-1.6.9.jar	HIGH	3	Highest	42
xmlsec-1.5.3.jar	MEDIUM	2	Highest	44
xstream-1.4.8.jar	HIGH	2	Highest	44
acionador-batch-negocio				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30
commons-beanutils-1.9.2.jar	HIGH	1	Highest	37
guava-14.0.1.jar	MEDIUM	1	Highest	21
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
jackson-databind-2.1.4.jar	CRITICAL	39	Highest	38
poi-3.13.jar	HIGH	4	Highest	29
weld-core-1.1.28.Final.jar	MEDIUM	1		27
slf4j-ext-1.7.2.jar	CRITICAL	1	Highest	29

módulos de negocio				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30
commons-beanutils-1.9.2.jar	HIGH	1	Highest	37
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
hibernate-search-orm-4.6.0.Final-redhat-2.	MEDIUM	1	Highest	34
solr-core-3.6.2.jar	HIGH	12		28
solr-solrj-3.6.2.jar	HIGH	10		27
spring-core-4.1.0.RELEASE.jar	CRITICAL	8	Highest	28
jackson-databind-2.1.4.jar	CRITICAL	39	Highest	38
poi-3.13.jar	HIGH	4	Highest	29
guava-14.0.1.jar	MEDIUM	1	Highest	21
weld-core-1.1.28.Final.jar	MEDIUM	1		27
slf4j-ext-1.7.2.jar	CRITICAL	1	Highest	29
axis-1.4.jar	HIGH	4	Highest	15

módulos rest				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
spring-core-4.1.0.RELEASE.jar	CRITICAL	8	Highest	28
axis-1.4.jar	HIGH	4	Highest	15
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
poi-3.13.jar	HIGH	4	Highest	29
logback-core-1.1.3.jar	CRITICAL	1	Highest	32
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30
resteasy-jaxrs-2.3.10.Final.jar	HIGH	2		21
weld-core-1.1.28.Final.jar	MEDIUM	1		27
slf4j-ext-1.7.2.jar	CRITICAL	1	Highest	29
guava-14.0.1.jar	MEDIUM	1	Highest	21
jasperreports-6.1.1.jar	HIGH	5	Low	29
commons-beanutils-1.9.2.jar	HIGH	1	Highest	37
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
bcprov-jdk14-138.jar	HIGH	16	Highest	23
jackson-databind-2.1.4.jar	CRITICAL	39	Highest	38

auditoria-core				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-3.13.jar	HIGH	4	Highest	29
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
jackson-databind-2.5.4.jar	CRITICAL	40	Highest	40
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30

servicos-soap				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-3.13.jar	HIGH	4	Highest	29
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
jackson-databind-2.5.4.jar	CRITICAL	40	Highest	40
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30
log4j-1.2.17.jar	CRITICAL	2	Highest	29
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
commons-beanutils-1.9.2.jar	HIGH	1	Highest	37
wss4j-1.6.9.jar	HIGH	3	Highest	42
xmlsec-1.5.3.jar	MEDIUM	2	Highest	44
opensaml-2.5.1-1.jar	HIGH	5	Highest	37
xstream-1.4.8.jar	HIGH	2	Highest	44
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
cxf-rt-core-2.7.14.jar	HIGH	8	Highest	41
agendamento-negocio				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30
commons-beanutils-1.9.2.jar	HIGH	1	Highest	37
hibernate-search-orm-4.6.0.Final-redhat-2	MEDIUM	1	Highest	34
solr-core-3.6.2.jar	HIGH	12		28
solr-solrj-3.6.2.jar	HIGH	10		27
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
jackson-databind-2.1.4.jar	CRITICAL	39	Highest	38
poi-3.13.jar	HIGH	4	Highest	29
guava-14.0.1.jar	MEDIUM	1	Highest	21
weld-core-1.1.28.Final.jar	MEDIUM	1		27
slf4j-ext-1.7.2.jar	CRITICAL	1	Highest	29
negocio-core				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-3.13.jar	HIGH	4	Highest	29
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
jackson-databind-2.5.4.jar	CRITICAL	40	Highest	40
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
hibernate-search-orm-4.6.0.Final-redhat-2	MEDIUM	1	Highest	34
solr-core-3.6.2.jar	HIGH	12		28
solr-solrj-3.6.2.jar	HIGH	10		27
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30

servicos-core				
Dependency	**Highest Severity**	**CVE Count**	**Confidence**	**Evidence Count**
tika-core-1.4.jar	CRITICAL	10	Highest	30
pdfbox-1.8.1.jar	HIGH	3	Highest	25
jempbox-1.8.1.jar	HIGH	3	Highest	27
bcprov-jdk15-1.45.jar	Unknown	15	Highest	27
poi-3.13.jar	HIGH	4	Highest	29
poi-scratchpad-3.9.jar	HIGH	7	Highest	27
xmpcore-5.1.2.jar	HIGH	1		33
xercesImpl-2.8.1.jar	Unknown	2	Low	67
jackson-databind-2.5.4.jar	CRITICAL	40	Highest	40
commons-collections-3.2.1.jar	CRITICAL	3	Highest	35
cdi-api-1.0-SP4.jar	HIGH	1	Low	30
log4j-1.2.14.jar	CRITICAL	2	Highest	23
commons-collections4-4.0.jar	HIGH	1	Highest	37
jackson-mapper-asl-1.9.9.jar	CRITICAL	14	High	30

A planilha acima apresenta as vulnerabilidades encontradas nas dependências de cada módulo, o detalhamento encontra-se no Anexo I deste documento.

Há certa recorrência no relatório de vulnerabilidade de dependências entre os componentes do sistema, estas estão correlacionadas a arquitetura de referencia e necessitam ser revistas.

4.3 OWASP ZAP

A ferramenta funciona como scanner de segurança, utilizada para realização de testes de vulnerabilidade de aplicações WEB. Atualmente trata-se de um dos projetos mais ativos na comunidade de software livre.

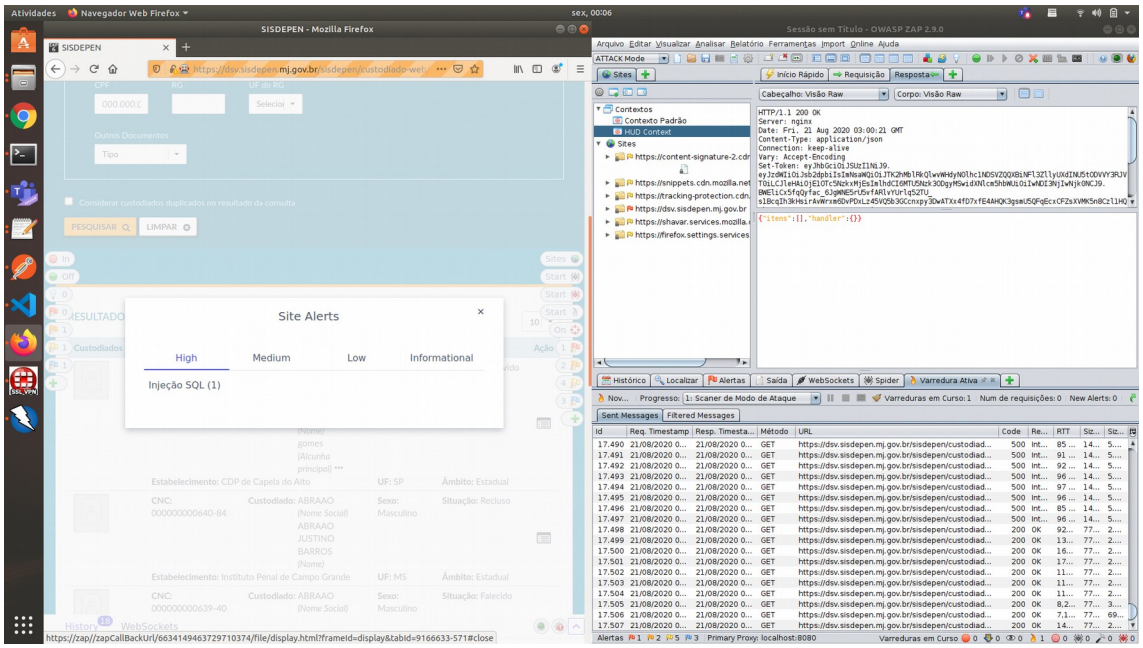

Figura 16: Análise de vulnerabilidade

O relatório completo deste teste está disponível no anexo I deste documento, o detalhamento desta análise está classificada em:

- 1 vulnerabilidade de severidade alta;
- 2 vulnerabilidades de severidade média;
- 10 vulnerabilidades de baixa média;
- 6 vulnerabilidades a nível informativo;

4.4 Análise sobre os resultados

Este tópico tratará tecnicamente a análise baseado nos resultados obtidos pelas ferramentas citadas juntamente com a análise amostral do código fonte.

4.4.1 Manutenibilidade de código

Os relatórios apresentados pela ferramenta SonarQube demonstram uma série de vícios adotados durante o processo de construção do software que necessitam de revisão e atribuem certa vulnerabilidade e instabilidade de comportamento da solução.

A existência de cobertura de testes de unidade na camada de serviço da aplicação traz consigo a facilidade no processo de refactoring e manutenção da aplicação, uma vez que há condições de mensurar impactos durante o processo de manutenção corretiva/adaptativa.

A aplicação apresenta boa estrutura arquitetural e padronização de código aliado a baixa complexidade ciclomática e a boa coesão, fatores estes que facilitam a manutenção do código.

4.4.2 Confiabilidade

Existe o controle transacional a nível de aplicação utilizando especificação JTA, este é o tratamento segue boas práticas de desenvolvimento de aplicações sendo esta a camada responsável por orquestrar as execuções em banco de dados. Este controle transacional garante as propriedades ACID do SGBD.

É importante ressaltar os resultados apresentados nos relatórios das ferramentas de análise de vulnerabilidade de dependências e análise de vulnerabilidade da aplicação, estes pontos precisam de

atenção principalmente no tocante a vulnerabilidade por injeção de SQL, sendo que esta compromete os dados do sistema.

4.4.3 Performance e estabilidade

Não foi analisado a aplicação em funcionamento para avaliar demais requisitos não funcionais. Durante o processo de análise de código fonte não fora encontrado evidências que demonstrem impactos em performance da aplicação.

4.4.3 Escalabilidade

A arquitetura baseada em módulos por serviços e o comportamento sem estado (stateless) da aplicação back-end promove boa capacidade de escalonamento na horizontal com a utilização de cluster e balanceadores de carga.

Esta arquitetura além de promover ambiente escalonável, favorece a utilização de ambientes redundantes com maior probabilidade de tolerância a falhas.

5 Recomendações

É altamente recomendado que seja efetuado refactoring de código dos bugs e vulnerabilidades de código apontadas pelo SonarQube , estas atividades certamente trarão maior confiabilidade a ferramenta e estabilidade em seu uso. Para os demais itens apontados pela ferramenta SonarQube durante o processo de análise de código são altamente desejáveis, contudo este processo de ajuste de código é moroso, contudo, minimizado pela boa cobertura de testes de unidade.

Ajustar as dependências que trazem maior risco para a aplicação é altamente recomendável, recomenda-se que este trabalho deve ser feito de forma analítica e cautelosa afim de não prejudicar a estabilidade da ferramenta. Sugere-se a interseção das vulnerabilidades apresentadas pelas ferramentas OWASP ZAP para que sejam associadas e corrigidas. Esta recomendação esta embasada na interseção de resultados das ferramentas utilizadas e na otimização e na assertividade do trabalho de refactoring.

Recomenda-se também que seja instalado o agente da ferramenta de APM do Ministério da Justiça nos ambientes de homologação e produção, criar métricas e alarmes auxiliam na continuidade do serviço (monitoramento de processamento e memória por exemplo) tendo em vista que esta ferramenta fornece mecanismos para determinarmos o comportamento da solução (auxiliam no refactoring de código) também subsidia para o correto dimensionamento da infraestrutura.

6 Conclusão

A aplicação apresenta uma boa estruturação em sua construção o que facilita a sua manutenção corretiva/evolutiva, embora a versão do famework Angular utilizada na construção do frontend dos módulos esteja defasada, não há viabilidade técnica para o seu upgrade, contudo não há uma real dificuldade na manutenção corretiva/evolutia da solução.

Necessita-se que sejam removidos ou substituir as dependências de componentes de negócio do SERPRO e demais integrações que não o MJ não julgue como negocialmente necessárias, sendo elas: ferramenta Control-M (utilizada para agendamento de tarefas), componente SERPRO CAPTCHA, integração com o Login Único, rintegração com o SINESP/PPE, SERPRO/DNE e SERPRO/DNE.